

# MITTEILUNGSBLATT DES RECHENZENTRUMS DER UNIVERSITAET ERLANGEN-NUERNBERG



NR. 12 -ERLANGEN- AUGUST 1972

# RECHENZENTRUM

der Friedrich-Alexander-Universität Erlangen-Nürnberg  
8520 Erlangen, Martensstr. 1, Tel. 09131/852360

## Vorstand:

Prof. Dr. W. Händler  
Prof. Dr. W. Schüßler  
Prof. Dr. W. Specht

## Leiter:

Dr. F. Wolf

Herausgegeben vom Rechenzentrum der Universität Erlangen-Nürnberg

---

Die Druckvorlage für das Titelblatt wurde mit Hilfe des an die Rechanlage CD 3300 angeschlossenen Zeichengerätes vom Typ CALCOMP(Incremental Plotter) gezeichnet.

MITTEILUNGSBLATT  
DES RECHENZENTRUMS DER  
UNIVERSITÄT ERLANGEN-NÜRNBERG

Digitized by the Internet Archive  
in 2026



---

---

# MITTEILUNGSBLATT DES RECHENZENTRUMS DER UNIVERSITAET ERLANGEN-NUERNBERG

---

---

GRAPHEMSYSTEM UND WORTKONSTITUENZ

DOKUMENTATION EINES LINGUISTISCHEN PROGRAMMSYSTEMS

JUERGEN SCHOENHUT

DISSERTATION AN DER PHILOSOPHISCHEN FAKULTAET DER  
FRIEDRICH-ALEXANDER-UNIVERSITAET ERLANGEN-NUERNBERG

D29

---

NR. 12 -ERLANGEN- AUGUST 1972

---

MITTEILUNGSBLATT  
DES RECHENZENTRALS DER  
UNIVERSITÄT ERLANGEN-NÜRNBERG

ERFORSCHUNGS- UND WIRTSCHAFTSLEHRE  
UNIVERSITÄT ERLANGEN-NÜRNBERG



LEIDEN BBL.

Druck von wissenschaftlichen Publikationen HOGEL, 8520 Erlangen, Hauptstr. 109

NR. 12 - ERLANGEN - AUGUST 1912

## V o r w o r t

An dieser Stelle ist es mir eine angenehme Pflicht, all jenen zu danken, die mir beim Entstehen dieser Arbeit mit ihrem sachkundigen Rat und ihrer Hilfsbereitschaft zur Seite standen.

Mein besonderer Dank gilt meinem Lehrer Professor Dr. Emil E. Ploss, der die Anregung gab, von einem historischen Text auszugehen und die Entwicklung der Methode mit Interesse verfolgte, Herrn Dr. Erich Straßner, der keine Mühe beim Austesten der Programme scheute und mit mir gemeinsam so manche Nacht und so manches Wochenende im Rechenzentrum ausharrte und stets zu sachlichen Diskussionen bereit war, Herrn Helmut Weinacht, der mit seinen Texten die Problematik bereicherte, Herrn Dr. Walther L. Fischer vom Mathematischen Institut, der bei allen Schwierigkeiten, insbesondere bei mathematischen Fragen, für uns immer ein offenes Ohr hatte, Herrn Professor Dr. Peter F. Ganz, der mich auf die Bibelübersetzung Ms.Bodl. 97o aufmerksam machte und mir einen Mikrofilm der Offenbarung dieser Handschrift besorgte, und Herrn Dr. Günther Bauer von der Handschriften-Abteilung der Universitätsbibliothek Erlangen für die Einsicht in den Bibeldruck von J.Mentel (Inc. 3o) und die Möglichkeit, Teile davon zu kopieren. Schließlich danke ich der Leitung, den Mitarbeitern und den Operateuren des Rechenzentrums der Friedrich-Alexander-Universität Erlangen-Nürnberg für ihre praktische Hilfsbereitschaft und den nicht selbstverständlichen freien Zugang zur Rechenanlage und Herrn Gerhard Koller für das zur Verfügung gestellte Worthäufigkeitsprogramm und für das Anfertigen der Mikrofilmkopie eines Textauschnittes für diese Arbeit. Herzlicher Dank gilt auch Herrn Romig vom Rechenzentrum, der die Tuschearbeiten an der Druckvorlage erledigte. Last not least ganz besonderen Dank meiner Frau Christel, die mir beim Ablochen viel Mühe ersparte und die Reinschrift dieser Arbeit anfertigte.



# Inhaltsverzeichnis

1. Einleitung
2. Allgemeine Vorbemerkungen
- 2.1 Der Bereich "Linguistische Datenverarbeitung" (LDV)
  - 2.1.1 Bisherige Anwendungen der LDV
  - 2.1.2 Versuch einer groben Gliederung der LDV
- 2.2 Überlegungen zu einem datenintensiven Programmsystem
- 2.3 Aufbau der Dokumentation
- 2.4 Metasprachliche Notation zur Beschreibung eines Programmsystems
- 2.5 Vereinbarungsteile und graphische Beschreibung
- 2.6 Implementierung
- 2.7 Verwendete Texte
3. Dokumentation
  - 3.1 Metasprachliche Beschreibung und Erläuterungen zu den problemlösenden Bestandteilen des Programmsystems
    - 3.1.01 Die Operation \*MCP[MS, CODE; LK]
      - 3.1.01.1 Exkurs 1: Codierung von historischen Texten auf maschinenlesbaren Datenträgern
    - 3.1.02 Die Operation LSTA[LK, LPAR; HIST, CHIQ]
      - 3.1.02.1 Exkurs 2: Signifikante Textmenge
    - 3.1.03 Die Operation \*UMF[P, CHIQ, FLAG; FLAG]
    - 3.1.04 Die Operation KONK[LK, SINF; KREG]
      - 3.1.04.1 Exkurs 3: Kontextregister
    - 3.1.05 Die Operation KMPW[KREG, RINF; KWF]
      - 3.1.05.1 Exkurs 4: Komprimierte Wortformendatei
    - 3.1.06 Die Operation FWB[LK; WH]
      - 3.1.06.1 Exkurs 5: Funktion von Worthäufigkeitslisten in linguistischer Textanalyse
    - 3.1.07 Die Operation PRAE[KWF, PRFX; KWFP]
      - 3.1.07.1 Exkurs 6: "Praefixe"
    - 3.1.08 Die Operation KETT[KWFP, ZTYP; TKET]
      - 3.1.08.1 Exkurs 7: Zeicheneinteilung in Klassen
    - 3.1.09 Die Operation KMPK[TKET, KINF; KTKT]

- 3.1.10 Die Operation UEMK [TKET;UEKT]
- 3.1.11 Die Operation \*SYS [UEKT,KWFP,KREG;VGRS]
- 3.1.11.1 Exkurs 8: Die Erkennung von /F/, /U/ und /W/,  
im Text repräsentiert als F, U, V oder W
- 3.1.12 Die Operation GRUL [VGRS;RULE,PRUL]
- 3.1.13 Die Operation TRNS [KWFP,RULE;GRWF,RSTA]
- 3.1.14 Die Operation \*KGS [GRWF,RSTA,KWFP,VGRS;VGRS]
- 3.1.15 Die Operation \*EXR [RULE,RSTA;RRUL]
- 3.1.16 Die Operation UEMG [GRWF,RRUL;UEGR,PUEG]
- 3.1.17 Die Operation KONX [UEGR,METR;KONM,PKON]
- 3.1.18 Die Operation CLSS [KONM,SCHW;NTPL,VNTP]
- 3.1.18.1 Exkurs 9: Algorithmus zur Berechnung der Schwell-  
werte
- 3.1.19 Die Operation \*KCS [NTPL,GRWF,RSTA,KWFP,KREG,  
VGRS;VGRS]
- 3.1.20 Die Operation WSEG [GRWF,RRUL;SEGW]
- 3.1.21 Die Operation WKST [SEGW,WKH,WKSH]
- 3.2 Beschreibung des Gesamtablaufs der Problemlösung
- 3.3 Kurze Beschreibung der verwendeten Datenwartungs-  
und Datensicherungsrouitinen
- 3.4 Ausführliche graphische Beschreibung der implemen-  
tierten problemlösenden Programme
- 3.4.01 Programmbeschreibung LSTA
- 3.4.02 Programmbeschreibung KONK
- 3.4.03 Programmbeschreibung KMPW
- 3.4.04 Programmbeschreibung FWB
- 3.4.05 Programmbeschreibung PRAE
- 3.4.06 Programmbeschreibung KETT
- 3.4.07 Programmbeschreibung KMPK
- 3.4.08 Programmbeschreibung UEMK
- 3.4.09 Programmbeschreibung GRUL
- 3.4.10 Programmbeschreibung TRNS
- 3.4.11 Programmbeschreibung UEMG
- 3.4.12 Programmbeschreibung für die Operatoren KONX  
und CLSS
- 3.4.13 Programmbeschreibung WSEG

- 3.4.14    Programmbeschreibung WKST
- 4.        Geplante und teilweise realisierte Erweiterungen
- 5.        Verzeichnis der Beispiel-Outputs
- 6.        Bibliographie
- 6.1      Abkürzungsverzeichnis zur Bibliographie



## 1. E i n l e i t u n g

Die vorliegende Untersuchung ist das Ergebnis etwa vierjähriger Entwicklungsarbeit. Der Weg führte von ersten Ansätzen in der Konkordanz-Erstellung bis hin zu einem lauffähigen Programmsystem für den Bereich "Graphemsystem und Wortkonstituenz". Dieses Programmpaket wurde inzwischen an umfangreichem Textmaterial getestet. Die entwickelten Algorithmen und Verfahren werden in dieser Arbeit in symbolischer Form beschrieben. Beispiellisten veranschaulichen die Outputs der jeweiligen Programme. Die Beschreibung wurde so angelegt, daß dieses System ohne große Umstellungsschwierigkeiten auch auf anderen Großrechenanlagen implementiert werden kann.

Die Überlegungen, die zu der Themenstellung führten, sollen hier kurz erwähnt werden. Einmal hat gerade die rege Diskussion der letzten Zeit in der Graphematik<sup>1</sup> dieses Arbeitsgebiet stark in den Vordergrund geschoben. Zum anderen wurde die Wahl einer datenverarbeitenden Methode mit bestimmt durch die umfangreichen Textcorpora, die in diesem Teilgebiet der linguistischen Beschäftigung mit historischen Texten der Aufarbeitung harren. Auch das Bestreben, in der empirischen Linguistik eine den Naturwissenschaften vergleichbare Methodik zu entwickeln, prädestiniert geradezu die Verwendung von Computern. Dadurch besteht eine reelle Chance, Fehler in Hypothesen durch empirische Überprüfung an umfangreichen Texten zu finden und gezielt diese Fehler zu korrigieren. Natürlich stand am Anfang der Untersuchungen auch die Frage, inwieweit syntaktische Analysen an historischem Textmaterial automatisch durchgeführt werden können. Dazu war es aber wiederum nötig, wollte man keine normalisierten Texte verwenden, ein Verfahren für eine automatische Reduzierung unterschiedlicher Schreibungen gleicher Wortformen zu entwickeln,

um nicht in der syntaktischen Analyse beim Wörterbuch zu scheitern. Insofern versteht sich die vorliegende Arbeit als Vorstufe zu einer automatischen syntaktischen Analyse von Texten historischer Sprachstufen (ohne genormte Rechtschreibung). Die hier aufgezeigten Methoden wurden an frühneuhochdeutschen Texten entwickelt und erprobt; prinzipiell bestehen jedoch keine Schwierigkeiten, diese Verfahren auf andere Sprachstufen auszu-dehnen.

---

<sup>1</sup>Vgl. etwa die Diskussion des Graphembegriffs bei:  
Harweg, R., Buchstabe und Graphem, in: LB 13, 1971, S.78-80.  
Wenzel, G., Beitrag zu Definition der Grapheme, in: LB 13, 1971, S.80-81.  
Piirainen, I.T., Grapheme als quantitative Größen, in: LB 13, 1971, S.81-82.  
Singer, H., Der Graphembegriff bei der Analyse alt-deutscher Handschriften, in: LB 13, 1971, S.83-85.  
Hakkarainen, H.J., Graphematik und Philologie, in: LiLi 1/2, 1971, S.191-204.

## 2. Allgemeine Vorbemerkungen

### 2.1 Der Bereich "Linguistische Datenverarbeitung" (LDV)

#### 2.1.1 Bisherige Anwendung der LDV

Das Arbeitsgebiet der Linguistischen Datenverarbeitung, wie es sich bisher darbot, war ein sehr heterogener Bereich, der allein durch den Einsatz elektronischer Großrechenanlagen zusammengehalten wurde<sup>1</sup>. Die ersten und bislang auch breitest-gestreuten Anwendungen lagen im Bereich der Index- und Konkordanz-Erstellung. Hier gibt es schon eine große Zahl von Einzelpublikationen und Reihen<sup>2</sup>. Von dieser Entwicklung profitierte schon frühzeitig auch das Arbeitsgebiet der Lexikographie<sup>3</sup>. Ein weiterer, relativ bekannt gewordener Einsatz von Computern liegt im Bereich der automatischen syntaktischen Analyse<sup>4</sup>. Einige dieser Versuche tendierten in Richtung auf automatische Übersetzung<sup>5</sup> von einer Sprache in eine andere; befriedigende Ergebnisse konnten bislang jedoch nur bei starken Einschränkungen erzielt werden, sei es in Bezug auf das Textcorpus, auf das die Analyse anwendbar war, oder in Bezug auf die erkannten Strukturen. In jüngster Zeit wird auch der Versuch unternommen, mit Hilfe von Dialogverfahren zuverlässige Analysealgorithmen zu gewinnen<sup>6</sup>. Schließlich wurden Datenverarbeitungsanlagen eingesetzt, um geschriebene Texte phonetisch zu transkribieren<sup>7</sup>. Weiter gab es Untersuchungen, die metrische Analysen von Texten automatisch durchführten. Die Versuche, die Technik der Edition zu automatisieren, stehen noch relativ weit am Anfang. Durch die Fortschritte im Bereich der elektronischen Syntax-Analyse hat sich neuerdings das Interesse an der automatischen Erstellung lemmatisierter Wörterbücher<sup>9</sup>

stark vergrößert. Schließlich wird in letzter Zeit auch die Ebene der schriftlichen Fixierung von Texten als eigenständiger Objektbereich Graphematik anerkannt. So war es naheliegend, auch in diesem Bereich nach Ansätzen zur LDV zu suchen<sup>10</sup>. Die vorliegende Arbeit geht von diesem Bereich aus, um zu größeren linguistischen Einheiten im Bereich der Wortbildung zu gelangen.

---

<sup>1</sup>Der einzige mir bekannte Versuch, darüber hinaus ein theoretisches Konzept der LDV zu entwickeln, wurde am Institut für Kommunikationsforschung und Phonetik der Universität Bonn gemacht; cgl. dazu: Ungeheuer, G., Linguistische Datenverarbeitung - die Realität und eine Konzeption, in: IBM-Nachrichten 21, 1971, S.688-694.

<sup>2</sup>Stellvertretend für die vielen Publikationen seien hier drei Reihen genannt: Schmitt, L.E.(Hg), Deutsche Wortindices, Berlin 1970ff (bisher 3 Bände). Schwerte, H., Schanze, H.(Hg), Indices zur deutschen Literatur, Frankfurt/Main 1968ff (bisher 7 Bände). Wisbey, R.(Hg), Compendia, Computer-Generated Aids to Literary and Linguistic Research, Leeds 1968ff (bisher 3 Bände).

<sup>3</sup>Vgl. dazu etwa: de Tollenaere, F., Die Automatisierung in der Lexikographie, in: BLI 1, 1963, S.33-44. ders., Nieuwe Wege in de Lexicologie, Amsterdam 1963.

<sup>4</sup>Vgl. etwa: Eggers, H.u.a., Elektronische Syntaxanalyse der deutschen Gegenwartssprache, Tübingen 1969. von Held, W., Eine verallgemeinerte Phrasenstrukturgrammatik mit einem Satzanalyse-Algorithmus, IPK-F 33, 1971.

<sup>5</sup>Etwa das sogenannte Georgetown-Projekt, das jetzt wieder neu aufgegriffen wurde; dazu: Janda, J.W., Rothkegel, A., Zimmermann, H., Dokumentation eines Programms zur Analyse russischer Sätze; Teil 1, in: BLI 19, 1970, S.7-39; Teil 2, in: BLI 20, 1970, S.7-31.

<sup>6</sup>Ein Konzept zur dialogischen Syntaxanalyse mit dem Computer befindet sich im Deutschen Seminar, Erlangen, gerade in der Entwicklung.

<sup>7</sup>Vgl. etwa:

Kučera, H., Mechanical phonemic transcription and phoneme count of Czech, IJSLP 6, 1963, S.36-50.

Ungeheuer, G., Kaestner, W., Untersuchung zu Transformation deutscher Schrifttexte in entsprechende Phonemtexte mit Hilfe elektronischer Rechenmaschinen, IPK-F 66/1, 1966.

<sup>8</sup>Vgl. etwa die Referate beim Symposium über Probleme der maschinellen Verarbeitung mittelhochdeutscher Texte, Mannheim, 11.-12.6.1971, von: Kochendörfer, G., Theoretische Grundlagen für die Technifizierung der recensio. Simon, G., Zur Theorie der Kontamination.

<sup>9</sup>Vgl. dazu:

Rath, R., Vorschläge zur Automatischen Lemmatisierung (AL) deutscher Adjektive, in: LB 12, 1971, S.53-59.

Vgl. auch die Referate der germanistischen Arbeitsgruppe, in: Referate des 1. SFB-Kolloquiums vom 20.-22. Mai 1971, Bericht 10-71, Sonderforschungsbereich Elektronische Sprachforschung, Universität des Saarlandes, Saarbrücken, 1971.

<sup>10</sup>Der erste Schritt in diese Richtung wurde von I.T. Piirainen getan: Piirainen, I.T., Graphematische Untersuchungen zum Frühneuhochdeutschen, Berlin 1968.

### 2.1.2 Versuch einer groben Gliederung der LDV

Die LDV gliedert sich im wesentlichen in zwei große Teilaspekte, die man mit den Stichworten "Entwickeln und Testen von Algorithmen" und "Verarbeitung möglichst vieler und umfangreicher Texte mit Hilfe dieser Algorithmen" charakterisieren kann. Da diese Einteilung nicht von vornherein einsichtig und plausibel ist, sollen einige Erläuterungen hierzu gegeben werden.

Die Anforderungen, die an ein Programm (einen in einer Programmiersprache formulierten Algorithmus) im ersten Fall gestellt werden, beziehen sich hauptsächlich auf bequeme und einfache Änderungsmöglichkeiten sowie auf möglichst komfortable Testhilfen; die Laufzeiten der Programme sowie deren Betriebsmittelanforderungen (insbesondere Kernspeicher) spielen hier eine untergeordnete Rolle, es sei denn, daß dafür die Kapazität des Systems nicht ausreichen würde.

Bei der eigentlichen Datenverarbeitung, unserem zweiten Bereich, liegen die Forderungen anders; dort kommt es darauf an, daß die Anforderungen an das System, also Rechenzeit und Betriebsmittelanforderung minimiert werden. Da diese beiden Faktoren i.a. gegenläufige Tendenzen zeigen, - bei sinkender Rechenzeit wird etwa die Speicheranforderung größer und umgekehrt - muß im jeweiligen Einzelfall über deren Priorität entschieden werden, und zwar abhängig von Problemstellung (I/O-intensiv oder rechenintensiv), Anlagentyp, Konfiguration und Betriebssystemfähigkeiten.

Leider wird heute in den meisten Fällen diese Trennung nicht in ausreichendem Maße wahrgenommen. Die Hauptbeschäftigung liegt im ersten Bereich, der zweite wird als unwichtig abgetan.

Was David G.Hays 1962 in seinem Referat für die

NATO Summer School mit dem Thema "Automatic Translation of Languages" in Venedig beklagte, gilt heute noch immer:

".... it was hard to find much interest in linguistic research of the empirical sort. Two areas were far more attractive: the design and refinement of translation algorithms, and the establishment of mathematical theory for linguistics. Yet each algorithm either contains or presupposes a body of empirical fact which, in fact, does not presently exist, and theory is pertinent to linguistics and its applications only insofar as it guides the collection and organization of data."<sup>1</sup>

Der Schwerpunkt der vorliegenden Arbeit lag in beiden Bereichen gleichermaßen: einmal wurden Algorithmen entwickelt und getestet, dann wurden die zugehörigen Programme optimiert und schließlich an Hand von größeren Datensätzen getestet. Das zur Verfügung gestellte Programmsystem hat inzwischen seine Feuerprobe bestanden; bisher wurden damit schon über 10 verschiedene Texte ausgewertet. In diesen Arbeiten<sup>2</sup> wird eine ausführliche Interpretation der Ergebnisse zu finden sein. Dies auch noch für die hier verwendeten Beispieltex te zu leisten, hätte den Rahmen dieser Arbeit gesprengt. Sie soll vielmehr die Methoden und Verfahren zugänglich machen.

---

<sup>1</sup> Hays, D.G., An Introduction to Computational Procedures in Linguistic Research, in: Ghizetti, A. (Hg), Automatic Translation of Languages. Papers presented at NATO Summer School held in Venice, July 1962, Oxford-London 1966, S.139-165; Zitat S.139.

<sup>2</sup> Straßner, E., Klassenbildung auf der Graphemebene und Wortkonstituenz, in Vorbereitung.  
Weinacht, H., Das Ambraser Heldenbuch (Arbeitstitel Diss.), in Vorbereitung.

## 2.2 Überlegungen zu einem datenintensiven Programmsystem

Ein Programmsystem wie das im folgenden beschriebene ist, wie fast alle Produktionsprogramme im Bereich der linguistischen Datenverarbeitung, stark datenintensiv. Angesichts des großen Aufwandes, der für die Datenerfassung nötig ist, müssen neben den eigentlichen problemlösenden Bestandteilen die Belange der Datenpflege und -sicherung eine dominierende Rolle spielen. Bei der Prioritätsbeurteilung der Datenwartung sollte neben dem manuellen Aufwand auch die benötigte Maschinenkapazität zur Erstellung dieser Daten berücksichtigt werden. Aus den angeführten Gründen versteht es sich, daß zu jedem umfangreichen System ein Paket von leistungsfähigen Datenwartungsroutinen zur Verfügung stehen muß. Solche Routinen sollten sowohl sicher als auch leicht zu handhaben sein.

Ein zweiter wichtiger Punkt ist die Programmwartung. Da das Austesten von komplexen Programmsystemen kein endliches Problem ist, muß auf eine gute Wartungsmöglichkeit besonders geachtet werden. Dazu ist es wichtig, eine ausführliche Dokumentation aller Teile zu besitzen. Dies für das zu beschreibende Programmsystem zu leisten, ist ein wesentliches Ziel dieser Arbeit.

### 2.3 Aufbau der Dokumentation

An eine gute Dokumentation eines Programmsystems aus dem Bereich der LDV müssen folgende Forderungen gestellt werden:

1. Die zugrundeliegenden Algorithmen müssen vollständig dargestellt sein.
2. Die Darstellung der Algorithmen und des Programmaufbaus sollte nicht zu sehr von der Anlage abhängig sein, auf der das Programmsystem implementiert wurde, - damit scheiden Assemblerprogramm-Listen aus.
3. Es sollte leicht möglich sein, von der Dokumentation ausgehend, das Programmsystem auf einer anderen Anlage zu implementieren.

Aus diesen Anforderungen ergeben sich Konsequenzen in der Darstellung:

1. Zur Darstellung der Makrostruktur des Programmsystems wurde eine besondere Notation entwickelt.
2. Algorithmen werden in Form von graphischen Beschreibungen angegeben, die auch alle nötigen organisatorischen Teile enthalten; die einzelnen Anweisungen werden symbolisch in einer FORTRAN<sup>1</sup> - Notation mit erweitertem Sprachumfang beschrieben.
3. Zu jedem Programm wird ein FORTRAN-Vereinbarungsteil beigegeben, auf den sich die symbolischen Notationen in den Flußdiagrammen beziehen.

---

<sup>1</sup>FORTRAN, weitestverbreitete Programmiersprache, Abkürzung von FORMula TRANslator; die in dieser Arbeit verwendete Version von FORTRAN ist eine Erweiterung der von United States of America Standards Institute festgelegten Normsprache USASI X3.9 FORTRAN, vgl. auch: USA Standard FORTRAN, USAS X3.9-1966, New York 1966.

## 2.4 Metasprachliche Notation zur Beschreibung eines Programmsystems

Wir fassen Programme als Operatoren auf, die sich auf Operanden (Dateien und/oder Parameter) beziehen. Bei den Operanden unterscheiden wir zwei Arten: solche, die zum Eingabevektor und solche, die zum Ausgabevektor eines Operators gehören. Diese Einteilung gilt immer nur in Bezug auf den betrachteten Operator, d.h. ein Operand, der beim einen Operator im Ausgabevektor steht, kann bei einem anderen Operator durchaus im Eingabevektor stehen. Für Operatoren wie für Operanden benutzen wir symbolische Namen. Im folgenden verwenden wir zur Definition der Syntax der Metasprache die Backus-Naur-Notation, wie sie im "Revised Report on the Algorithmic Language ALGOL 60"<sup>1</sup> verwendet wird.

### Definition der Metasprache

#### 1. Operationen

```

<Operation>      ::= <Operator>[<Eingabevektor>;
                        <Ausgabevektor>]
<Eingabevektor> ::= <Dateiname>|<Dateiname>,
                        <Eingabevektor>
<Ausgabevektor> ::= <Dateiname>|<Dateiname>,
                        <Ausgabevektor>
<Operator>       ::= <C1>|<C1><C>|<C1><C><C>|
                        <C1><C><C><C>
<Dateiname>      ::= <C>|<C><C>|<C><C><C>|<C><C><C><C>
<C>              ::= <Buchstabe>|<Ziffer>
<C1>            ::= <Buchstabe>|*
<Buchstabe>      ::= |A|B|C|D|E|F|G|H|I|J|K|L|M|N|
                        O|P|Q|R|S|T|U|V|W|X|Y|Z
<Ziffer>         ::= 0|1|2|3|4|5|6|7|8|9
    
```

Die mit \* beginnenden Operatoren bezeichnen manuelle Eingriffe. Taucht derselbe Dateiname sowohl im Eingabe- als auch im Ausgabevektor auf, so hat die Operation nach Ausführung den Wert 'wahr', wenn an der Datei Änderungen vorgenommen wurden; in allen anderen Fällen hat sie den Wert 'falsch'. Dieser Wert wird im folgenden verschiedentlich auch Funktionswert der Operationen genannt.

## 2. Ablaufkontrolle

Durch folgende Spracherweiterung ist es dann möglich, Sprünge und bedingte Sprünge auszuführen und damit die Möglichkeit einer Rückkopplung vorzusehen.

```
<Anweisung>      ::= <Operation>.|<Sprungziel>:
                   <Operation>.|<Sprunganweisung>.|
                   <Sprungziel>:<Sprunganweisung>.|
                   <Bedingung>.|<Sprungziel>:
                   <Bedingung>.
<Sprungziel>      ::= <Ziffer>|<Ziffer><Sprungziel>
<Sprunganweisung> ::= GOTO <Sprungziel>| STOP
<Bedingung>       ::= IF(<Operation>)<Operation>|
                   IF(<Operation>)<Sprunganweisung>
```

Der ausführbare Teil eines Programmsystems wird dann als lineare Abfolge von Anweisungen betrachtet, die durch bedingte und unbedingte Sprünge durchbrochen wird und deren logisches Ende durch STOP gekennzeichnet ist.

---

<sup>1</sup>Naur, P.(Hg), Revised Report on the Algorithmic Language ALGOL 60, in: Numerische Mathematik 4, 1963, S.420-453.

## 2.5 Vereinbarungsteile und graphische Beschreibung

Vereinbarungsteile sowie graphische Beschreibungen der Programme werden durch ein eigenes für diese Dokumentation entwickeltes Programmsystem<sup>1</sup> gewonnen. Der Vereinbarungsteil eines jeden Teilprogrammes enthält erst die Liste sämtlicher Einsprungpunkte in das Programm (PROGRAM-Anweisung, FUNCTION- bzw. SUBROUTINE-Anweisung und ggf. sämtliche ENTRY-Anweisungen dieses Programms); danach folgen sämtliche Vereinbarungen des Programms (INTEGER, REAL, CHARACTER, LOGICAL, DIMENSION, EQUIVALENCE, COMMON, EXTERNAL, DATA etc.); abgeschlossen wird diese Liste mit der Zusammenstellung sämtlicher in diesem Programm vorkommenden FORMAT-Anweisungen.

In der graphischen Beschreibung sind Formate und Vereinbarungsanweisungen eliminiert. Die verwendeten graphischen Zeichen sind nicht die der Flußdiagramm-Norm<sup>2</sup>, da diese Symbole von Form und Größe her nicht geeignet sind, ausreichend Text aufzunehmen. Die lineare Abfolge im Programm wurde auch bei der graphischen Beschreibung des Programmablaufs beibehalten, da dies in FORTRAN an gewissen Stellen nötig ist (Reihenfolge bei DO-Schleifen). Anfangs- und Endpunkte im Ablauf werden durch ovale Figuren dargestellt. Der Anfang von DO-Schleifen (die DO-Anweisung) wurde als Kasten mit Spitzen links und rechts dargestellt. Sprunganweisungen (IF, GOTO) werden durch Kästchen dargestellt, die eine nach rechts weisende Spitze haben. Normale Sprunganweisungen werden durch Pfeile auf der rechten Seite der Anweisungskästchen dargestellt. Alle übrigen Anweisungen werden in rechteckigen Kästchen untergebracht. Die Kontrolle bei DO-Schleifen wird durch gestrichelte Pfeile auf der linken Seite der Anweisungskästchen dargestellt. Sie gehen ggf. auch über mehrere Seitengrenzen hinweg. Bei normalen Sprunganweisungen, bei denen der Sprung über die Seiten-

grenze hinweg geht, wird nur ein Pfeil gezeichnet, der an der Pfeilspitze die Anweisungsnummer des Sprungziels und nach einem Schrägstrich die Seite, auf der diese Anweisungsnummer definiert ist, stehen hat. Beim Sprungziel steht entsprechend ein Pfeil nach innen und am Rand die Anweisungsnummer, nach einem Schrägstrich gefolgt von allen Seitenzahlen, von denen aus diese Anweisung angesprungen wird. Wird eine Anweisung von einer Seite aus mehrfach angesprungen, so wird die Seite entsprechend oft aufgeführt.

- 
- <sup>1</sup> Dazu:  
Schönhut, J., Ein komfortables Dokumentationssystem für FORTRAN-Programme, demnächst in: Mitteilungsblatt des Rechenzentrums der Universität Erlangen-Nürnberg.
- <sup>2</sup> DIN 66001 Informationsverarbeitung, Sinnbilder für Datenfluß- und Programmablaufpläne, 1966.

## 2.6 Implementierung

Das hier vorgelegte Programmsystem wurde auf der Rechenanlage des Rechenzentrums der Universität Erlangen-Nürnberg vom Typ Control Data CD 3300 unter dem Betriebssystem MASTER implementiert<sup>1</sup>. Als Implementierungssprachen wurden eine Erweiterung der Sprache FORTRAN<sup>2</sup>, der Assembler der Anlage, COMPASS<sup>3</sup> und die am Rechenzentrum von C. Endres entwickelte Mischsprache MIX-FORTRAN<sup>4</sup> verwendet. Die problemlösenden Teile wurden mit Ausnahme einiger Standard-Unterprogramme in FORTRAN geschrieben.

Im folgenden seien die Spracherweiterungen, die der von der Firma zur Verfügung gestellte FORTRAN-Übersetzer verarbeitet, kurz skizziert.

Hierzu gehört insbesondere der direkte Zugriff zu einzelnen Schriftzeichen im Kernspeicher. Für diesen Zweck wurde der spezielle Variablentyp CHARACTER eingeführt.

Weiterhin gibt es die Möglichkeit, beim Anstoßen der Ausführung eines Programms an das Hauptprogramm Parameter zu übergeben. Der Parameterempfangsbereich wird im Hauptprogramm in der ersten Anweisung, der PROGRAM-Anweisung (nicht Standard), in Parameterklammern hinter dem Programmnamen angegeben.

In Unterprogrammen gibt es über den herkömmlichen Einsprungpunkt hinaus weitere Einsprungpunkte zu definieren durch sogenannte ENTRY-Anweisungen. Sie enthalten keine Parameterliste, sondern verwenden die Liste, die in der SUBROUTINE- oder FUNCTION-Anweisung angegeben ist. Als Variable, der der Funktionswert in einem FUNCTION-Unterprogramm zugewiesen werden muß, bleibt der Name in der FUNCTION-Anweisung.

Zusätzlich zu herkömmlichen Hollerith-Konstanten kann man Zeichenketten in DATA- und FORMAT-Anweisungen zwischen Stern oder Apostroph eingeschlossen

darstellen. Für die Länge der Hollerith-Konstanten gilt keine Einschränkung, wenn sie in DATA-, FORMAT- oder CALL-Anweisungen verwendet wird. Eine weitere Darstellungsart von Zeichenkonstanten ist die sogenannte R-Konstante, die wie eine Hollerith-Konstante aufgebaut ist, die Zeichen jedoch rechtsbündig im Speicher enthält und links mit Nullen auffüllt.

Beim Kompilieren kann man wählen zwischen der Möglichkeit, standardmäßig für INTEGER-Variablen den gleichen Speicherplatz zu belegen, wie für REAL (in diesem Fall zwei Maschinenworte zu je 24 bits), oder unter der sogenannten S-Option (Single Word Integer Option) für INTEGER nur je ein Maschinenwort zu reservieren. Bei Verwendung der S-Option ist dies in der Dokumentation durch eine Kommentarkarte angegeben.

Ebenfalls nicht standardmäßig ist eine besondere Adressierungsart für Variablen und Felder, der sogenannte Chapter 2 Common oder Numbered Common.

Variablennamen dürfen bis zu acht Zeichen lang sein, im Gegensatz zur Norm, wo sie nur maximal sechs Zeichen lang sein können.

In dem vorhandenen FORTRAN-Compiler gibt es außerdem noch einige zusätzliche Anweisungen zu Ein-Ausgabe und zur internen Datenübertragung. Es sind dies Ein-Ausgabe auf den Standardgeräten (Kartenleser Kanal 60, Schnelldrucker Kanal 61) READ f,liste und PRINT f,liste, wobei f die Formatangabe und liste die Liste der Variablen bedeutet, weiter Anweisungen für gepufferte Ein-Ausgabe BUFFER IN (k,p)(anf, end) und BUFFER OUT (k,p)(anf,end), wobei k die Kanalnummer, p eine Paritätsangabe (o bedeutet BCD, 1 bedeutet binär), anf die erste und end die letzte auszugebende Variable eines zusammenhängenden Blockes im Kernspeicher ist. Statusabfragen sind möglich über die Funktionen IFEOF(k) und IFUNIT(k) ( für gepufferte Ein-Ausgabe). Wenn die Operation ordnungsgemäß beendet

wurde, liefern diese Funktionen den Wert o. Zur internen Datentransfer im Kernspeicher dienen die Anweisungen ENCODE (c,f,r)list (Übertragung und Formatierung gemäß Format f der Variablenliste list in den Bereich r wobei nach c Zeichen in r ein neuer "Record" begonnen wird) und DECODE (c,f,r)list (Umkehroperation zu ENCODE).

In den graphischen Beschreibungen wurde auch eine FORTRAN-Notation mit diesem erweiterten Sprachumfang benutzt. Die bei der Implementierung verwendeten selbstgeschriebenen Standard-Unterprogramme werden, sofern sie in der Dokumentation vorkommen, kurz in ihrer Funktion beschrieben.

- 
- <sup>1</sup> Zur Konfiguration der Rechananlage des Rechenzentrums siehe:  
Mitteilungsblatt des Rechenzentrums der Universität Erlangen-Nürnberg, Nr.10, Erlangen, 1972, S.5-8.  
Zur Rechananlage CD 3300 vgl.:  
3300 Computer System Reference Manual, Control Data Corporation, Pub.No.60157000, 1968<sup>3</sup>.  
Zum Betriebssystem MASTER vgl.:  
3300 3500 Computer Systems MASTER Version 3 Reference Manual, Control Data Corporation, Pub.No.60281000, 1970.
  - <sup>2</sup> 3100 3200 3300 3500 Computer Systems USASI FORTRAN MASTER MSOS Reference Manual, Control Data Corporation, Pub.No.60281400, 1970.
  - <sup>3</sup> 3100 3200 3300 3500 Computer Systems COMPASS Reference Manual, Control Data Corporation, Pub.No. 60236800A, 1969.
  - <sup>4</sup> Endres, C.- MIX-FORTRAN, vervielfältigte Beschreibung, demnächst in: Mitteilungsblatt des Rechenzentrums der Universität Erlangen-Nürnberg.

## 2.7 Verwendete Texte

Die in dieser Arbeit gegebenen Listen-Beispiele beziehen sich auf zwei Texte, die im folgenden jeweils abgekürzt angeführt werden:

1. Offenbarung der frühneuhochdeutschen Bibelübersetzung in der Handschrift Ms. Bodl. 97o der Bodleian Library, Oxford 316a - 322d, abgekürzt als BODL<sup>1</sup>.
2. Offenbarung der frühneuhochdeutschen Bibelübersetzung in dem Bibeldruck von Johann Mentel, ca. 1466, nach dem Exemplar der Universitätsbibliothek Erlangen, Inc. 3o, 395d - 4ood, abgekürzt als MENT.

---

<sup>1</sup>Auf diese Bibelhandschrift wurde ich von Professor Dr. Peter F. Ganz, Oxford, aufmerksam gemacht; er hat mir freundlicherweise auch einen Mikrofilm dieses Abschnittes besorgt.

### 3. Dokumentation

#### 3.1 Metasprachliche Beschreibung und Erläuterungen zu den problemlösenden Bestandteilen des Programmsystems

##### 3.1.01 Die Operation \*MCP[MS, CODE; LK] =====

\*MCP bezeichnet das Codieren und Lochen des Textes nach dem Original (Handschrift, Druck). \*MCP hat zwei Eingabeparameter: den Text MS, wie er in der Handschrift vorliegt und die Codetabelle CODE. Die Ausgabedatei LK befindet sich auf Lochkarten.

CODE enthält sämtliche Schriftzeichen, die in dem zu codierenden Text vorkommen, und die dazugehörigen Lochkartencodes. Tabelle 1 enthält z.B. die Codiervorschriften für MENT und BODL (mit Angabe der Druckerrepräsentation).

Außerdem gilt eine 1:1-Korrespondenz zwischen Zeile in MS und Dateneinheit (Lochkarte) in LK. Zur Kennzeichnung des Beginns einer neuen Spalte, Seite, Texts werden besondere Dateneinheiten in LK verwendet, die die Information darüber tragen.

Beispiel 1 zeigt einen Ausschnitt aus der Handschrift (BODL) und seine Codierung für die Maschine.

Tabelle 1: Codetabelle (CODE) für die Texte MENT und BODL

| MENT/BODL                           | Lochkartenbeschriftung | Drucker  |
|-------------------------------------|------------------------|----------|
| a/A                                 | A                      | A        |
| b/B                                 | B                      | B        |
| c/C                                 | C                      | C        |
| d/D                                 | D                      | D        |
| e/E                                 | E                      | E        |
| f/F                                 | F                      | F        |
| g/G                                 | G                      | G        |
| h/H                                 | H                      | H        |
| i/I                                 | I                      | I        |
| j/J                                 | J                      | J        |
| k/K                                 | K                      | K        |
| l/L                                 | L                      | L        |
| m/M                                 | M                      | M        |
| n/N                                 | N                      | N        |
| o/O                                 | O                      | O        |
| p/P                                 | P                      | P        |
| q/Q                                 | Q                      | Q        |
| r/R                                 | R                      | R        |
| s/S                                 | S                      | S        |
| ſ/S                                 | S                      | S        |
| ß                                   | S ]                    | S†       |
| t/T                                 | T                      | T        |
| u/U                                 | U                      | U        |
| v/V                                 | V                      | V        |
| w/W                                 | W                      | W        |
| x/X                                 | X                      | X        |
| y/Y                                 | Y                      | Y        |
| z/Z                                 | Z                      | Z        |
| 3/3                                 | Z ]                    | Z†       |
| ä/Ä                                 | A=                     | A=       |
| ö/Ö                                 | O=                     | O=       |
| ü/Ü                                 | U=                     | U=       |
| æ/Æ                                 | AE+                    | AE+      |
| ø/Ø                                 | OE+                    | OE+      |
| œ/Œ                                 | UE+                    | UE+      |
| n̄                                  | N/                     | N/       |
| (Kürzel z.B.<br>bzw. Unleserliches) | <DER>                  | ¬DER‡    |
| epi tel<br>(Fremdwort)              | 10 EPISTEL             | †EPISTEL |
| ihe us<br>(Name)                    | :IHESUS                | <IHESUS  |
| XXVI<br>(römische Zahlen)           | 26                     | 26       |

BEISPIEL 1.1

=====

32 NACH DYSEM \$AH ICH VIR ENGEL \$TEN  
33 AUFF DEN VIR ECKEN "DER% ERDEN DY ENT-  
34 HIELTEN DY VIR WIND "DER% ERDE/ DAS "DER% WIND  
35 WARDT WEHET NIT AUFF DI ERDEN/ NOCH AUFF  
36 DAS MERE NOCH IN KEINEN PAUM VND  
37 JCH \$JOHANNES \$AH EINE/ ANDERN ENGEL AB  
38 \$TEIGEN UO/ DEM AUF GANG "DER% \$UNNE/ HAB-  
39 END DAS Z+EICHEN DEZ+ LEBENDEN GOTZ+ VND  
40 \$CHREY MIT GROSSER \$TYM/E DEN VIR ENG-  
41 ELN DEN DO IST GEGEBEN Z+U \$CHADE/ DER  
42 ERDEN VND DEM MERE \$PRECHENT JR SULT  
43 NIT \$CHADEN DER ERDEN NOCH DEM MERE  
44 NOCH DEN PAUME/ VNCZ+ ALS LANG VNCZ+  
45 DAS WIR BEGZ+EICHEN DY KNECHT VN\$ERS  
46 GOTZ+ IN IR \$TYRN VND ICH HORT DY Z+AL  
47 "DER% GECZ+EICHENTEN/ HUNDERT TAU\$ENT VN/  
48 VIRVNOVIRCZ+IK TAU\$ENT GECZ+EICHENT

BEISPIEL 1.2

=====

**N**ach dysem sah ich vir engel \$ten  
auff den vir ecken d'erden dy ent-  
hielten dy virwind d'erde das d'wind  
wardt wehet nit auff di erden noch auff  
das mere noch in keinem paum / und  
Ich Johannes sah eine andern engel ab-  
steigen us dem auf gang d'sunne hab-  
end das zeichen der lebenden gotz und  
schrey mit grosser styme den vir eng-  
eln den do ist gegeben zu schade der  
erden und den mere sprachent jr sult  
nit schaden der erden noch dem mere  
noch den paume uns als lang uns  
das wir begreiffen dy knecht vnser  
gotz in ir sturn und ich hort dy sal  
d'gequachten hundert tausent  
vir und vnzik tausent gequachten

### 3.1.01.1 Exkurs 1: Codierung von historischen Texten auf maschinenlesbaren Datenträgern

Die Probleme, die die Codierung historischer Texte betreffen, sind eng verknüpft mit dem Medium, das für die maschinenlesbare Speicherung der Texte verwendet wird. So hat z.B. der Lochstreifen (bes. 8-Kanal-Lochstreifen) heute wegen seines (relativ) großen Zeichenvorrates vielfach bei der Aufbereitung von Texten Eingang gefunden. Das Korrigieren von einmal erstellten Texten ist verhältnismäßig aufwendig. Die zweite weit verbreitete Möglichkeit ist eine Texterfassung auf Lochkarte; dort steht nur ein Zeichenvorrat von maximal 64 Zeichen zur Verfügung. Das zwingt in einigen Fällen dazu, Mehrfachcodierungen vorzunehmen, d.h. ein Zeichen des Textes durch zwei oder mehr Zeichen auf der Lochkarte darzustellen. Korrekturen sind sehr leicht möglich. Aus diesem Grunde und da uns kein geeignetes Erfassungsgerät für Lochstreifen zur Verfügung stand, wählten wir als Datenträger die Lochkarte. An die dabei nötige Mehrfachcodierung gewöhnt man sich sehr schnell.

In letzter Zeit haben wir mit gutem Erfolg versucht, auch Datenerfassung am Display (Daten-sichtgerät, Bildschirm mit Tastatur) durchzuführen. Die Fehlerquote scheint bei diesem Verfahren erheblich niedriger zu liegen, als beim vergleichbaren Schreiben von Lochkarten.

Die Frage, was codiert werden muß, richtet sich letztlich nach dem Verwendungszweck. In unseren Fällen haben wir etwa in Stichproben überprüft, inwieweit die Großschreibung von Interesse ist. Das Ergebnis war negativ und hat die Datenerfassung etwas erleichtert, zumal im Lochkartencode keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht wird.

### 3.1.o2 Die Operation LSTA[LK,LPAR;HIST,CHIQ]

=====

LSTA ermittelt die Häufigkeit und die Übergangshäufigkeit von Zeichen in LK abhängig von der Zahl der berücksichtigten Zeichen. LPAR gibt die Anfangszahl der Zeichen und die Schrittweite an, bei der jeweils der aktuelle Stand der Statistik aufgezeichnet wird. Man kann damit die Skalen der Textlänge variieren.

HIST ist eine dreiteilige Datei; sie enthält im ersten Teil die aufgezeichneten statistischen Werte, die Information von LPAR und ggf. nicht berücksichtigte Teile von LK, im zweiten graphische Darstellungen für Übergangshäufigkeit als Funktion der Länge für jeden Zeichenübergang bezüglich des ersten und des zweiten Zeichens, und im dritten Teil die relative Häufigkeit der Zeichen abhängig von der Textlänge, ebenfalls in graphischer Form.

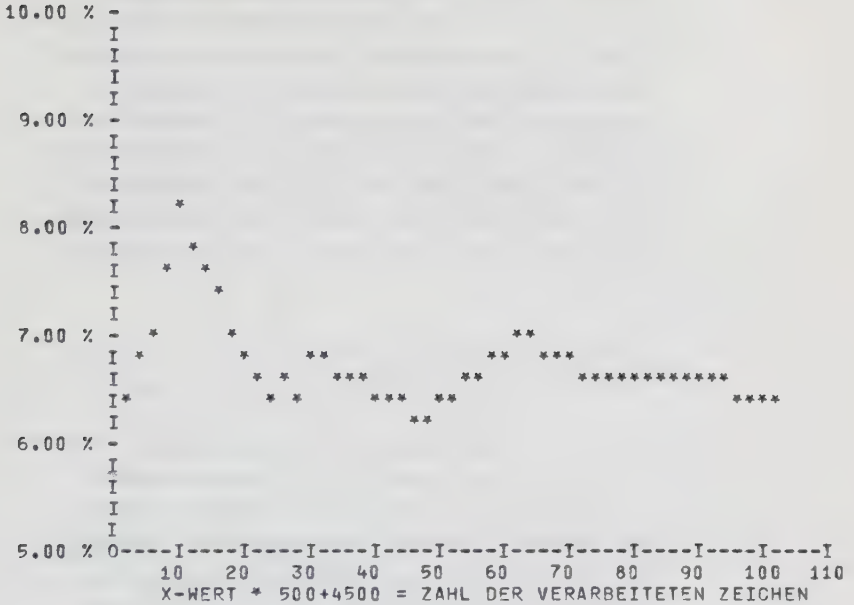
HIST gibt schon einen Überblick über die statistische Vertrauenswürdigkeit der ausgewählten Textstichprobe; die relativen Häufigkeiten pendeln sich mit zunehmender Textlänge auf einen Mittelwert ein. CHIQ enthält die Zahl der Freiheitsgrade und den Wert für  $\chi^2$ .

Beispiel 2 (BODL) möge zur Veranschaulichung dienen.

# BEISPIEL 2

=====

RELATIVE HAEUFIGKEIT VON A VOR N ALS FUNKTION  
DER TEXTLAENGE (ZAHL DER ZEICHEN)  
100 % = ZAHL DER N



### 3.1.02.1 Exkurs 2: Signifikante Textmenge

Bei jeder Untersuchung, die sich, wie die Analyse von Texten, auf Stichproben beschränken muß, steht am Anfang die Frage nach dem nötigen Stichprobenumfang, oder mit anderen Worten: Sicherheit der Ergebnisse und Aufwand müssen in einer vernünftigen Relation zueinander stehen. Im Rahmen der Statistik wurden Signifikanztests entwickelt, die inzwischen auch in der quantitativen Linguistik<sup>1</sup> ihren festen Platz haben. Hier spielt etwa die statistische Meßzahl 'Standardabweichung' eine Rolle. Sie wird gewonnen nach der Formel

$$\sigma = \sqrt{pq}/N$$

wobei  $p$  die Häufigkeit eines Ereignisses [%],  $q$  die Differenz  $100 - p$  und  $N$  die Gesamtzahl der beobachteten Ereignisse. Gemäß dem Gesetz von Bernoulli besteht eine Wahrscheinlichkeit von ca.  $1/3$ , daß die Abweichung mehr als den einfachen Standardfehler beträgt, eine Wahrscheinlichkeit von ca.  $1/20$  mehr als der doppelten und eine Wahrscheinlichkeit von ca.  $1/370$ , daß sie mehr als den dreifachen Standardfehler beträgt. Man kann nun Stichproben vergleichen und die Differenz der zugehörigen  $p$  bilden. Der Quotient  $\frac{p_1 - p_2}{\sigma}$  gilt dann als Signifikanzkriterium; ist der Wert kleiner als 2, dann hat man es mit 95% Sicherheit ( $1 - 1/20$ ) mit zufälligen Stichproben aus einer Gesamtheit zu tun.

Eine vergleichbare Rolle zur Standardabweichung bei Einzelereignissen spielt der sogenannte  $\chi^2$ -Test bei der Signifikanzbewertung aller unterschiedlichen Ereignisse (Zeicheninventar). Führt die Standardabweichung zu ebensovielen Ergebnissen, wie Zeichen im Zeichenvorrat waren, so liefert der  $\chi^2$ -Test ein Ergebnis für die Gesamtheit der Zeichen.

Angenommen, wir haben eine Matrix mit den Häufigkeiten  $f_{ij}$  Zeichen  $j$  in den verschiedenen Stichproben  $i$ .  $n_j$  sei die Summe aller Zeichen  $j$  in allen Stichproben, also:

$$n_j = \sum_{i=1}^m f_{ij},$$

wobei  $m$  die Zahl der Stichproben ist.  $N_i$  ist die Zahl aller Zeichen in Stichprobe  $i$ , also

$$N_i = \sum_{j=1}^p f_{ij},$$

wobei  $p$  die Größe des Zeicheninventars ist.  $N$  sei gegeben durch die Summe aller  $N_i$  bzw.  $n_j$ , also

$$N = \sum_{i=1}^m N_i = \sum_{j=1}^p n_j.$$

Unter diesen Bedingungen und unter der Voraussetzung, daß die relative Häufigkeit eines Zeichens unabhängig von der Stichprobe sei, bekommt man für jeden  $f_{ij}$  einen Erwartungswert  $f'_{ij}$  mit

$$f'_{ij} = n_j N_i / N.$$

Man hat also eine Vergleichsmöglichkeit durch die Differenz  $f_{ij} - f'_{ij}$ , relativiert durch Division durch  $f'_{ij}$ ; aus theoretischen Gründen bildet man das Quadrat dieser Differenz  $(f_{ij} - f'_{ij})^2 / f'_{ij}$ .

Der Wert von  $\chi^2$  berechnet sich dann aus der Formel

$$\chi^2 = \sum (f_{ij} - f'_{ij})^2 / f'_{ij}.$$

Für diesen Wert gibt es Vergleichstabellen<sup>2</sup>; dabei benötigt man die Zahl der sogenannten Freiheitsgrade, die sich aus  $(m-1)(p-1)$  berechnet und die festgelegte Fehlerwahrscheinlichkeit  $P$ . Ein für die Praxis ausreichender und meist benutzter Wert für  $P$  ist 0.05, d.h. 5% Fehlerwahrscheinlichkeit.

---

<sup>1</sup>Das Standardwerk hierzu:

Herdan, G., Quantitative Linguistics, London 1964. Die folgenden Ausführungen beziehen sich weitgehend auf dieses Werk, S.32-40.

<sup>2</sup>Tabelle bei Herdan, a.a.O., S.39; ausführliche Tabellen finden sich in:  
Geigy, I.R.(Hg), Wissenschaftliche Tabellen, Biberach 1955.

### 3.1.03 Die Operation \*UMF[P,CHIQ,FLAG;FLAG]

=====

Der Operator \*UMF ermittelt, ob die Textlänge ausreichend ist, um innerhalb der vorgegebenen Fehlerwahrscheinlichkeit P vertrauenswürdige Aussagen zu machen. Ist der Textumfang zu gering, so wird der Parameter FLAG umgesetzt, die Operation liefert also nach Ausführung den Funktionswert 'wahr'.

Ist der Text innerhalb der gegebenen Einschränkungen ausreichend lang, so bleibt der Parameter FLAG unverändert, der Funktionswert nach Ausführung der Operation ist also 'falsch'.

### 3.1.04 Die Operation KONK[LK,SINF;KREG]

=====

Mit Hilfe von KONK gewinnt man aus dem in LK vorliegenden Text ein Kontextregister KREG; SINF enthält die dazu nötige Steuerinformation, etwa welche Zeichen als Wortgrenze fungieren sollen, welche maximale Wortlänge angenommen werden soll, etc. SINF ist abhängig von den gewählten Codierungen in CODE. KREG enthält sämtliche im laufenden Text vorkommenden Wortformen mit Längenangaben, Zitatenachweis und Belegzeile (Lochkarte) in alphabetischer Reihenfolge.

Beispiel 3 ist ein Ausschnitt aus KREG in dem Text BODL.

#### BEISPIEL 3

=====

|               |         |                                                    |
|---------------|---------|----------------------------------------------------|
| 3 TYR         | 3210/06 | MIT \$EINEM GE\$IND VND IR TYR "ODER% DY           |
| 3 TYR         | 3210/34 | TYR VND \$EIN PILD NOCH NIT \$EIN Z+EICHE/         |
| 3 TYR         | 322A/03 | FEWERS VND \$WEFELS DO AUCH DAS TYR                |
| 5 TYRE/       | 3190/34 | DEM \$TUL GOTZ+ VND UOR DEN VIR TYRE/              |
| 5 TYRE/       | 3208/21 | \$NU=REN VND EINS AUS DEN VIR TYRE/ GAB DE/        |
| 6 TYREN/      | 317C/24 | GELN VND ICH HORT EINS UON DEN VIR TYREN/          |
| 5 TYRES       | 317C/46 | ICH DY \$TYM/E DEZ+ VIROE/ TYRES \$PRECHEN         |
| 4 TYRS        | 319C/54 | VND EZ+ TET ALLEN DEN GEWALT DEZ+ ER\$TE/ TYRS.    |
| 4 TYRS        | 320C/13 | MUND DEZ+ TYRS VND UON DEM MUND DEZ+               |
| 4 TYRS        | 3210/10 | HETEN DAS Z+EICHEN DES TYRS VND DY                 |
| 6 UAL\$CH     | 3210/07 | ^BE\$TYA IST BEGRYFFEN VND MIT IR "DER% UAL\$CH    |
| 8 UAL\$CHEN   | 320C/14 | UAL\$CHEN "PPHETE/% AUS GEN DREY VNREINE           |
| 8 UAL\$CHEN   | 322A/04 | VND DY UAL\$CHEN "PPHETE/% GECEUCZ+IGTE            |
| 5 UASZ+       | 3208/22 | \$IBEN ENGELN \$IBEN GULOEIN UASZ+ VOL DEZ+ Z+ORN/ |
| 3 UAS         | 3208/32 | GEL VND GO\$Z+ AUS \$EIN UAS IN DI ERDE/ VND       |
| 4 UBER        | 317C/30 | UBER UBERWUND VND DO ER AUF GETET "DZ+%            |
| 4 UBER        | 317C/50 | WALT GEGEBEN UBER DI VIR TYR "DER% ERDEN Z+U       |
| 4 UBER        | 318C/18 | NEN VND HABEN UBER SICH EINE/ KUNIG EINE/          |
| 10 UBERWINDET | 316C/56 | UBERWINDET VND BEHU=T PIS IN "DZ+% END MEI/        |
| 10 UBERWINDET | 3200/40 | VND DAS LAMP UBERWINDET \$I WANN ER                |
| 8 UBERWUND    | 317C/30 | UBER UBERWUND VND DO ER AUF GETET "DZ+%            |
| 10 UBERWUNDEN | 3198/48 | VND \$I UBERWUNDEN IN DURCH "DZ+% PLUT DEZ+        |
| 8 UECHTENT    | 3200/39 | "DER% ^BE\$TIEN. DY\$E UECHTENT MIT DEM LAMB       |
| 4 UELT        | 318A/40 | NOCH DY \$UNN UELT AUFF \$YE NOCH KEIN             |
| 4 UE\$T       | 322B/31 | DY GRUNT UE\$T "DER% "MAWERX "DER% \$TAT AUS ALLER |
| 3 UIL         | 3168/05 | UIL WA\$\$ER VND HETT IN "SEINER% RECHTE/ HANT     |
| 3 UIL         | 3168/09 | "IRER% CRAFT VND DO JCH IN GESAH DO UIL ICH        |
| 3 UIL         | 3178/33 | NOCH GE\$EHEN IN ES VND ICH WEINET UIL WAN/        |
| 3 UIL         | 313A/53 | VA\$Z+ VND IM \$INT GEBE/ UIL ENCZ+UNDUNG          |

### 3.1.04.1 Exkurs 3: Kontextregister

Die Verwendung von maschinell erstellten Indizes ist heute schon selbstverständlich geworden. Die vielen Publikationen (vgl. 2.1.1, Anm.2) auf diesem Gebiet geben davon ein eindrucksvolles Bild.

Man kann dabei grundsätzlich zwischen drei verschiedenen Formen unterscheiden:

1. reine Wortformenbücher, die den Wortlaut und sämtliche Zitatstellen angeben,
2. sogenannte lemmatisierte Wörterbücher, in denen die Wortformen aus 1. in mühsamer Handarbeit - abgesehen von neueren Versuchen (vgl. 2.1.1, Anm.9), auch hier eine Teilautomatisierung zu erreichen - unter das jeweilige Lemma geordnet wurden, und
3. sogenannte Kontextregister, die zu jeder Wortform das Zitat und einen gewissen Kontext zu jedem Beleg bieten.

Für Arbeiten an Texten, deren Register nicht publiziert werden sollen, scheint die dritte Möglichkeit die günstigste zu sein. Die Erstellung solcher Kontextregister auf einen Computer ist heute eine triviale Angelegenheit. Zudem bedarf es keiner weiteren linguistischen Vorarbeiten wie etwa der Lemmatisierung. Auf der anderen Seite bieten sie gegenüber 1. den Vorteil, daß sich die Wortformen (meist) ohne Schwierigkeiten klassifizieren lassen, was sonst nur zusammen mit einer Ausgabe zu machen wäre. Insgesamt ist jedenfalls das Kontextregister ein gutes Arbeitsmittel, das mit nur geringem Aufwand verbunden ist.

### 3.1.05 Die Operation KMPW[KREG,RINF;KWF]

=====

KMPW komprimiert die in KREG mehrfach vorkommenden Wortformen zu einer Wortform mit Häufigkeitsangabe. In RINF steht die maximal verarbeitete Länge einer Wortform und eine Textkennung. Die Datei KWF enthält neben Wortlaut und Häufigkeit eine Dateikennung für den Text und die Art, die Wortlänge und eine fortlaufende Nummerierung.

In Beispiel 4 ist der entsprechende Ausschnitt zu Beispiel 3 (Operator KONK) enthalten.

#### BEISPIEL 4

=====

|    |            |          |    |      |
|----|------------|----------|----|------|
| 3  | TYR        | 800L.970 | 23 | 1837 |
| 5  | TYRE/      | 800L.970 | 2  | 1838 |
| 6  | TYREN/     | 800L.970 | 1  | 1839 |
| 5  | TYRES      | 800L.970 | 1  | 1840 |
| 4  | TYRS       | 800L.970 | 3  | 1841 |
| 6  | UAL\$CH    | 800L.970 | 1  | 1842 |
| 8  | UAL\$CHEN  | 800L.970 | 2  | 1843 |
| 5  | UA\$Z+     | 800L.970 | 1  | 1844 |
| 3  | UAS        | 800L.970 | 1  | 1845 |
| 4  | UBER       | 800L.970 | 2  | 1846 |
| 8  | UBER\$ICH  | 800L.970 | 1  | 1847 |
| 10 | UBERWINDET | 800L.970 | 2  | 1848 |
| 8  | UBERWUND   | 800L.970 | 1  | 1849 |
| 10 | UBERWUNDEN | 800L.970 | 1  | 1850 |
| 8  | UECHTENT   | 800L.970 | 1  | 1851 |
| 4  | UELT       | 800L.970 | 1  | 1852 |
| 4  | UE\$T      | 800L.970 | 1  | 1853 |
| 3  | UIL        | 800L.970 | 1  | 1854 |

#### 3.1.05.1 Exkurs 4: Komprimierte Wortformendatei

Bei Untersuchungen, bei denen es primär nicht auf die Reihenfolge irgendwelcher Einheiten im laufenden Text ankommt, empfiehlt es sich, die Daten vorher anders zu organisieren. Dies bedeutet insbesondere bei mehrfach vorkommenden Einheiten eine erhebliche Einsparung von Rechenleistung und Speicherkapazität. Diese Gesichtspunkte haben zwar in der linguistischen Datenverarbeitung bislang kaum Eingang gefunden. Dennoch scheinen sie eine vernünftige und mögliche, wenn man den Ausdruck der zu bearbeitenden Probleme mit einbezieht, wohl bald eine nötige Beschränkung zu sein. So kann man beispielsweise anstelle einer mehrfachen gleichartigen Verarbeitung der gleichen Wortform die einmalige Abarbeitung einer Wortform vorsehen, ggf. unter Berücksichtigung der Häufigkeit.

Als Ausgangsbasis für Untersuchungen an Einheiten bis zur Größe einer Wortform ist dies eine einfache und wirksame Methode zur Optimierung von Datensätzen.

### 3.1.06 Die Operation FWB[LK;WH]

=====

Der Operator FWB erstellt aus dem in LK vorliegenden Text eine nach der Häufigkeit geordnete Liste WH aller vorkommenden Wortformen. Innerhalb der gleichen Häufigkeit sind die Wortformen alphabetisch geordnet. Die Häufigkeitsangabe erfolgt in absoluten Zählwerten und in Prozent. Am Ende dieser Liste steht die Angabe der Zahl der laufenden und der verschiedenen Wortformen.

Beispiel 5 enthält die 20 häufigsten Wortformen in BODL.

#### BEISPIEL 5

=====

|     |       |      |       |
|-----|-------|------|-------|
| 1.  | VND   | 1100 | 9.602 |
| 2.  | DER%  | 370  | 3.230 |
| 3.  | IN    | 330  | 2.881 |
| 4.  | DY    | 261  | 2.278 |
| 5.  | DEM   | 236  | 2.060 |
| 6.  | DI    | 216  | 1.885 |
| 7.  | DO    | 185  | 1.615 |
| 8.  | DER   | 179  | 1.563 |
| 9.  | DAS   | 172  | 1.501 |
| 10. | I\$T  | 172  | 1.501 |
| 11. | DEN   | 152  | 1.327 |
| 12. | EIN   | 151  | 1.318 |
| 13. | ICH   | 148  | 1.292 |
| 14. | Z+U   | 111  | 0.969 |
| 15. | \$INT | 104  | 0.908 |
| 16. | \$Y   | 104  | 0.908 |
| 17. | MIT   | 97   | 0.847 |
| 18. | DEZ+  | 95   | 0.829 |
| 19. | UON   | 83   | 0.725 |
| 20. | ALS   | 80   | 0.693 |

BEI 11456 LAUFENDEN WORTFORMEN KAMEN 2208 VERSCHIEDENE VOR.

3.1.06.1 Exkurs 5: Funktion von Worthäufigkeitslisten in  
linguistischer Textanalyse

Listen der Wortformen, nach ihrer Häufigkeit geordnet, haben in den meisten Fällen nur eine unterstützende Funktion. Man wird sie in vielen Fällen bei der schwierigen Abgrenzung von zentralen und peripheren Erscheinungen zu Hilfe nehmen. Für Untersuchungen über den Wortschatz sowie bei der Ermittlung der Kompetenz aus der Performanz leisten solche Register gute Dienste. Gewöhnlich wird man sie aber im Zusammenhang mit alphabetischen Registern verwenden um den Zugriff zum Kontext zu ermöglichen.

### 3.1.07 Die Operation PRAE[KWF,PRFX;KWFP]

PRAE kennzeichnet die Grenze von "Praefix" (vgl. 3.1.07.1) und Restwortform; in PRFX steht, welches "Praefix" von welchen Wortformen abgespalten werden soll. Die Ergebnisdatei KWFP hat dieselbe Struktur wie KWF; einige Wortformen sind mit Praefixkennzeichnung versehen d.h., sie enthalten zwischen "Praefix" und Restwortform das Zeichen \* eingeschoben und sind dann um ein Zeichen länger.

Beispiel 6 zeigt den gleichen Ausschnitt aus KWF und KWFP.

#### BEISPIEL 6.1

|    |              |      |   |     |
|----|--------------|------|---|-----|
| 9  | BESPRENGT    | BOOL | 1 | 356 |
| 6  | BESTAT       | BOOL | 1 | 357 |
| 6  | BESTEN       | BOOL | 1 | 358 |
| 12 | BESUNDERLICH | BOOL | 1 | 359 |
| 8  | BETRACHT     | BOOL | 1 | 360 |
| 5  | BEWEG        | BOOL | 1 | 361 |
| 7  | BEWEGET      | BOOL | 1 | 362 |
| 6  | BEWEGT       | BOOL | 1 | 363 |
| 8  | BEWEGUNG     | BOOL | 1 | 364 |
| 8  | BESLOZT      | BOOL | 1 | 365 |

#### BEISPIEL 6.2

|    |              |      |   |     |
|----|--------------|------|---|-----|
| 10 | BE*\$PRENGT  | BOOL | 1 | 356 |
| 7  | BE*\$TAT     | BOOL | 1 | 357 |
| 7  | BE*\$TEN     | BOOL | 1 | 358 |
| 12 | BESUNDERLICH | BOOL | 1 | 359 |
| 9  | BE*TRACHT    | BOOL | 1 | 360 |
| 6  | BE*WEG       | BOOL | 1 | 361 |
| 8  | BE*WEGET     | BOOL | 1 | 362 |
| 7  | BE*WEGT      | BOOL | 1 | 363 |
| 9  | BE*WEGUNG    | BOOL | 1 | 364 |
| 9  | BE*\$LOZT    | BOOL | 1 | 365 |

### 3.1.07.1 Exkurs 6: "Praefixe"

Beim Erarbeiten von Kriterien zur sinnvollen Abgrenzung von sprachlichen Einheiten in einem gegebenen Text stellt man fest, daß Grenzen in manchen Fällen durch größere sprachliche Einheiten bestimmt werden. Um nun auf einer Stufe sprachlicher Zeichen nicht die gesamte(n) nächsthöhere(n) Abstraktionsstufe(n) als Ballast mit-schleppen zu müssen, scheint es sinnvoll, an den wenigen in Frage kommenden Stellen korrigierende Eingriffe vorzunehmen. Es wäre außerordentlich aufwendig und unökonomisch - wenngleich möglich -, zur Abgrenzung von Graphemen ein umfangreiches Stammwörterbuch und ein vollständiges System von morphologischen Regeln zur Verfügung zu stellen.

Da der Umfang der nötigen Korrekturen minimal ist, wurde dieses Verfahren gewählt; in vielen Fällen ist diese Korrektur vorbeugender Natur, um schon auf dieser Stufe Fehler in späteren Verarbeitungsschritten zu vermeiden.

### 3.1.08 Die Operation KETT[KWFP,ZTYP;TKET]

=====

In ZTYP ist die Menge aller verwendeten Zeichen in Klassen unterteilt. Die Klassen 1 ( $T_1$ ) und 2 ( $T_2$ ) bilden die hier berücksichtigten Zeichen. KETT segmentiert nun die Elemente von KWFP derart, daß immer maximale Ketten von Zeichen der gleichen Klasse  $T_1$  oder  $T_2$  gebildet werden. Nicht in  $T_1$   $T_2$  enthaltene Zeichen wirken als Kettengrenzzeichen.

Diese Ketten bilden zusammen mit Textkennzeichen, Dateimerkmal, Länge, Belegwortform und Häufigkeit die Ausgabedatei TKET.

Beispiel 7 veranschaulicht die Struktur von TKET (MENT).

#### BEISPIEL 7

=====

|   |     |        |      |   |
|---|-----|--------|------|---|
| 1 |     |        |      | 1 |
| 1 | A   | ALL    | MENT | 1 |
| 2 | LL  | ALL    | MENT | 1 |
| 1 |     |        |      | 1 |
| 1 | A   | ALLE   | MENT | 7 |
| 2 | LL  | ALLE   | MENT | 7 |
| 1 | E   | ALLE   | MENT | 7 |
| 1 |     |        |      | 7 |
| 1 | A   | ALLEN  | MENT | 2 |
| 2 | LL  | ALLEN  | MENT | 2 |
| 1 | E   | ALLEN  | MENT | 2 |
| 1 | M   | ALLEN  | MENT | 2 |
| 1 |     |        |      | 2 |
| 1 | A   | ALLE/  | MENT | 1 |
| 2 | LL  | ALLE/  | MENT | 1 |
| 1 | E   | ALLE/  | MENT | 1 |
| 1 | /   | ALLE/  | MENT | 1 |
| 1 |     |        |      | 1 |
| 1 | A   | ALLEN  | MENT | 1 |
| 2 | LL  | ALLEN  | MENT | 1 |
| 1 | E   | ALLEN  | MENT | 1 |
| 1 | N   | ALLEN  | MENT | 1 |
| 1 |     |        |      | 1 |
| 1 | A   | ALLTEN | MENT | 3 |
| 3 | LLT | ALLTEN | MENT | 3 |
| 1 | E   | ALLTEN | MENT | 3 |
| 1 | N   | ALLTEN | MENT | 3 |
| 1 |     |        |      | 3 |
| 1 | A   | ALLUMB | MENT | 1 |
| 2 | LL  | ALLUMB | MENT | 1 |
| 1 | U   | ALLUMB | MENT | 1 |
| 2 | MB  | ALLUMB | MENT | 1 |
| 1 |     |        |      | 1 |

### 3.1.08.1 Exkurs 7: Zeicheneinteilung in Klassen

Die Einteilung der in einem Text vorkommenden Schriftzeichen in zwei zu berücksichtigende Klassen  $T_1$  und  $T_2$  hat auf dieser Stufe der Bearbeitung lediglich den Sinn, einen groben Überblick über die vorkommenden Schriftzeichenkombinationen zu verschaffen. Für die praktische Arbeit hat es sich dabei als recht brauchbar erwiesen, die Schriftzeichen, die im gesprochenen Neuhochdeutschen Vokalen entsprechen, zu einer Klasse  $T_1$  zusammenzufassen, die Konsonanten entsprechen, zu der zweiten Klasse  $T_2$ . Die so gewonnene Aufteilung erhebt selbstverständlich keinen Anspruch auf linguistische Relevanz, sie bietet jedoch für die weiteren Untersuchungen eine erhebliche Erleichterung, auch wenn in Fällen von Kontextabhängigkeit (vgl. U in UAND und STUND) "Fehler" in Kauf genommen werden müssen. =

### 3.1.09 Die Operation KMPK[TKET,KINF;KTKT]

=====

KMPK sortiert TKET und komprimiert die Einträge auf je einen für jede unterschiedliche Kette aus TKET. In KINF steht die maximale Kettenlänge und ein Textkennzeichen. KTKT enthält neben Kettenlängen, Ketten und Textsigle auch die Vorkommenshäufigkeit.

Beispiel 8 für KTKT bezieht sich auf den Text MENT.

#### BEISPIEL 8

=====

|   |       |      |   |      |    |
|---|-------|------|---|------|----|
| 2 | CH    | MENT | K | 236  | 11 |
| 3 | CH\$  | MENT | K | 1    | 12 |
| 4 | CH\$+ | MENT | K | 2    | 13 |
| 3 | CHS   | MENT | K | 3    | 14 |
| 3 | CHT   | MENT | K | 53   | 15 |
| 4 | CHTS  | MENT | K | 1    | 16 |
| 2 | CK    | MENT | K | 5    | 17 |
| 1 | D     | MENT | K | 793  | 18 |
| 2 | DL    | MENT | K | 1    | 19 |
| 2 | DM    | MENT | K | 1    | 20 |
| 2 | DR    | MENT | K | 4    | 21 |
| 2 | D\$   | MENT | K | 1    | 22 |
| 1 | E     | MENT | K | 1346 | 23 |
| 2 | EE    | MENT | K | 2    | 24 |
| 2 | EI    | MENT | K | 179  | 25 |
| 2 | EO    | MENT | K | 3    | 26 |
| 2 | EU    | MENT | K | 12   | 27 |
| 3 | EUA   | MENT | K | 6    | 28 |
| 3 | EU=   | MENT | K | 21   | 29 |
| 4 | EU=E  | MENT | K | 1    | 30 |
| 2 | EY    | MENT | K | 31   | 31 |
| 3 | EYE   | MENT | K | 1    | 32 |
| 1 | F     | MENT | K | 33   | 33 |
| 2 | FF    | MENT | K | 32   | 34 |
| 3 | FFG   | MENT | K | 5    | 35 |
| 3 | FFN   | MENT | K | 2    | 36 |
| 3 | FFT   | MENT | K | 2    | 37 |
| 4 | FFTH  | MENT | K | 2    | 38 |
| 4 | FFZ+  | MENT | K | 1    | 39 |

### 3.1.10 Die Operation UEMK[TKET;UEKT]

UEMK bildet aus TKET eine Übergangsmatrix der vor-  
kommenden Ketten. Die Datei UEKT besteht aus zwei  
Teilen: der erste Teil enthält nur die Übergänge  
und deren Häufigkeit, daneben eine Textkennung;  
der zweite Teil enthält dieselbe Information mit  
der Angabe sämtlicher dazu belegter Wortformen  
(vgl. dazu Beispiel 9).

#### BEISPIEL 9 =====

|        |    |               |      |   |
|--------|----|---------------|------|---|
| LB     |    | HALB          | MENT | 2 |
| LB     | E  | HALBE         | MENT | 1 |
| LB     | E  | KALBE         | MENT | 1 |
| LD     |    | GOLD          | MENT | 1 |
| LD     | E  | GULDEN        | MENT | 1 |
| LD     | EI | GULDEIN       | MENT | 3 |
| LD     | I  | GULDIN        | MENT | 3 |
| LGT    |    | NACHUOLGT     | MENT | 1 |
| LL     |    | ALL           | MENT | 1 |
| LL     | E  | ALLE          | MENT | 7 |
| LL     | E  | ALLEN         | MENT | 2 |
| LL     | E  | ALLE/         | MENT | 1 |
| LL     | E  | ALLEN         | MENT | 1 |
| LL     | E  | GEUALLEN      | MENT | 1 |
| LL     | E  | HELLE         | MENT | 2 |
| LL     | E  | KRISTALLEN    | MENT | 1 |
| LL     | U  | ALLUMB        | MENT | 1 |
| LL\$CH | A  | GESELL\$CHAFT | MENT | 1 |
| LLT    |    | DERFU=LLT     | MENT | 1 |
| LLT    |    | HELLT         | MENT | 1 |
| LLT    | E  | ALLTEN        | MENT | 3 |

### 3.1.11 Die Operation \*SYS[UEKT,KWFP,KREG;VGRS]

Der Operator \*SYS wird benutzt zur Bildung eines vorläufigen Graphemsystems VGRS; VGRS ist eine später zu modifizierende Arbeitshypothese, die durch Vergleich der Übergänge von Ketten UEKT, der komprimierten Wortformen KWFP und (zum Erkennen von Homographen) des Kontextregisters zu dem betreffenden Text gewonnen wird.

Theoretisch könnte diese Operation im Rahmen des Gesamtprogrammsystems unterbleiben; von der Sorgfalt dieser Arbeitsstufe hängt jedoch im wesentlichen Maße die Zahl der später nötigen Iterationsschritte ab.

Für VGRS gilt folgende Syntax:

```

<VGRS>                ::= <VGRS-Segment> |
                        <VGRS-Segment> <VGRS>
<VGRS-Segment>        ::= <Graphembeschreibung>
                        <Varianten>
<Graphembeschreibung> ::= <lesbare Form des
                        Graphems> <Codierung>
<Varianten>           ::= <Schreibvariantenangabe> | <Schreibvariantenangabe> <Varianten>
<Schreibvariantenangabe> ::= <Schreibvariante> |
                        <Schreibvariante>
                        <Stellungsangabe>
<Stellungsangabe>     ::= INITIAL | NOTINIT | FINAL
    
```

Dabei wird jede Graphembeschreibung und jede Schreibvariantenangabe auf eine neue Lochkarte geschrieben. Zwischen der lesbaren Form des Graphems und seiner Codierung sowie zwischen Schreibvarianten und Stellungsangabe stehen mindestens 8 Leerzeichen.

Die Angaben auf den Lochkarten beginnen in Spalte 4. Die Karten mit den Schreibvariantenangaben werden in Spalte 2 mit einer 1 gekennzeichnet.

Zusätzlich gilt die Konvention, daß Grapheme von Typ 1 mit Zeichen codiert werden, die eine Oktalverschlüsselung  $\leq 20_8$  haben; unter bestimmten Kontextbedingungen (vgl. 3.1.11.1) sind Graphemgruppen zu verschlüsseln.

Beispiel 10 zeigt einen Ausschnitt aus VGRS (BODL).

# BEISPIEL 10

=====

|   |      |   |         |
|---|------|---|---------|
|   | A    | 0 |         |
| 1 | A    |   |         |
|   | E    | 1 |         |
| 1 | E    |   |         |
| 1 | A=   |   |         |
|   | I    | 2 |         |
| 1 | I    |   |         |
| 1 | Y    |   |         |
| 1 | IE   |   | NOTINIT |
| 1 | Y=   |   |         |
| 1 | YE   |   | NOTINIT |
| 1 | J    |   |         |
| 1 | IO   |   | NOTINIT |
|   | O    | 3 |         |
| 1 | O    |   |         |
|   | U    | 4 |         |
| 1 | U    |   |         |
| 1 | V    |   |         |
| 1 | JU   |   |         |
| 1 | W    |   | FINAL   |
|   | O=   | 6 |         |
| 1 | OE+  |   |         |
| 1 | O=   |   |         |
|   | U=   | 7 |         |
| 1 | U=   |   |         |
| 1 | UE+  |   |         |
|   | AU   | 8 |         |
| 1 | AU   |   |         |
| 1 | AW   |   |         |
| 1 | AV   |   |         |
|   | EI   | 9 |         |
| 1 | EI   |   |         |
| 1 | AI   |   |         |
| 1 | EY   |   |         |
| 1 | AY   |   |         |
|   | EU   | = |         |
| 1 | EU   |   |         |
| 1 | EU=  |   |         |
| 1 | EW   |   |         |
| 1 | EW=  |   |         |
| 1 | AU=  |   |         |
| 1 | A=U  |   |         |
| 1 | EUE+ |   |         |
| 1 | EU=W |   |         |
| 1 | EUW  |   |         |
|   | UO   | + |         |
| 1 | UO+  |   |         |

3.1.11.1 Exkurs 8: Die Erkennung von /F/, /U/ und /W/,  
im Text repräsentiert als F, U, V  
oder W

Die Zuordnung der Schriftzeichen F, U, V und W zu einem der Grapheme /U/, /W/ bzw. /F/ erfordert einen in der Regel vom jeweiligen Text abhängigen und von Text zu Text verschiedenen Algorithmus. Das hier gegebene Beispiel bezieht sich auf den Text BODL.

Hierbei kann man zunächst feststellen, daß F immer zum Graphem /F/ gehört. Bei U, V und W muß man verschiedene Fälle unterscheiden. Wir wollen hier eine Liste aller vorkommenden Kombinationen aufstellen und daraus eine Untermenge des Gesamtregelsystems gewinnen, wobei dann die Kombinationen unberücksichtigt bleiben, die nicht zu den Graphemen /F/, /U/ bzw. /W/ gehören (/AU/, /EU/, etc.).

AU=,AU,AV,AW,A=U

EU=,EU,EWA,EWEI,EWE,EWI,EWO,EWU,EW=,EW,

FF,F

UA,UE,UI,UO,U=,U,VA,VEI,VE,VIE,VO,VYE,V,W

Bei dieser Liste ist natürlich schon eine Vorauswahl getroffen, die nur die für das Gesamtregelsystem relevanten Kombinationen beinhaltet. Daraus ergeben sich für die Grapheme /F/, /U/ und /W/ folgende Regeln (notiert in Form von Ersetzungsregeln, wie sie in der generativen und transformationellen Grammatik üblich ist<sup>1</sup>):

$$\begin{pmatrix} U \\ V \end{pmatrix} \begin{pmatrix} A \\ E \\ I \\ O \end{pmatrix} \longrightarrow /F// \begin{pmatrix} A \\ E \\ I \\ O \end{pmatrix} /$$

$$V \begin{pmatrix} EI \\ YE \end{pmatrix} \longrightarrow /F// \begin{pmatrix} EI \\ I \end{pmatrix} /$$

$$W \# \longrightarrow /U/$$

$$W \longrightarrow /W/$$

$$\begin{pmatrix} U \\ V \end{pmatrix} \longrightarrow /U/$$

$$\begin{pmatrix} F \\ FF \end{pmatrix} \longrightarrow /F/$$

Für die Anwendungsreihenfolge der Regeln gilt, daß immer die längere Regel angewendet werden muß, wenn mehr als eine Regel anwendbar ist.

Die Zahl der hier angegebenen Regeln erhöht sich noch, wenn eines der behandelten Grapheme als Kontext für ein anderes Graphem benötigt wird. Im fertigen Regelsystem für den Text BODL sind 38 Regeln enthalten, in denen die Schriftzeichen U, V oder W auftauchen. Insgesamt besteht das Regelsystem aus 139 Regeln.

---

<sup>1</sup>Siehe etwa:

Bach, E. An Introduction to Transformational Grammars, New York, Chicago, San Francisco 1966.

### 3.1.12 Die Operation GRUL[VGRS;RULE,PRUL]

=====

Der Operator GRUL erzeugt aus dem in VGRS vorliegenden Graphemsystem ein System erkennender Regeln, RULE, die es gestatten, eine Wortform genau nach Vorschrift in ihr graphemisches Äquivalent, in Zukunft immer als "Graphemschreibung" bezeichnet, zu Übersetzen. RULE ist sortiert in einer Form auf Magnetplatte gespeichert, die eine leichte Verwendbarkeit bei der Übersetzung ermöglicht. PRUL ist eine Druckausgabe zur Kontrolle der Codierung.

Beispiel 11 ist ein Auszug aus PRUL.

BEISPIEL 11  
=====

|         |    |           |                         |         |
|---------|----|-----------|-------------------------|---------|
| GRAPHEM | A  | VARIANTEN | A                       |         |
| GRAPHEM | E  | VARIANTEN | E<br>A=                 |         |
| GRAPHEM | I  | VARIANTEN | I<br>Y<br>IE<br>Y=      | NOTINIT |
|         |    |           | YE<br>J                 | NOTINIT |
|         |    |           | IO                      | NOTINIT |
| GRAPHEM | O  | VARIANTEN | O                       |         |
| GRAPHEM | U  | VARIANTEN | U<br>V<br>#U<br>UO<br>W | FINAL   |
| GRAPHEM | O= | VARIANTEN | OE+<br>O=               |         |
| GRAPHEM | U= | VARIANTEN | U=                      |         |
|         |    |           | UE+                     |         |
| GRAPHEM | AU | VARIANTEN | AU<br>AW<br>AV          |         |
| GRAPHEM | EI | VARIANTEN | EI<br>AI<br>EY<br>AY    |         |
| GRAPHEM | EU | VARIANTEN | EU<br>EU=               |         |
|         |    |           | EW<br>EW=               |         |
|         |    |           | AU=                     |         |
|         |    |           | A=U                     |         |
|         |    |           | EUE+                    |         |
|         |    |           | EU=W                    |         |
|         |    |           | EUM                     |         |

3.1.13 Die Operation TRNS[KWFP;RULE,GRWF,RSTA]  
=====

Der Operator TRNS übersetzt die in KWFP vorliegenden Wortformen gemäß der in RULE festgelegten Regeln in die Graphemschreibung GRWF. RSTA gibt Aufschluß über die Zahl der Anwendungen einer jeden Regel. Über RSTA kann man Einblick gewinnen in die Verteilung der Varianten eines Graphems sowie über die Relevanz der einzelnen Regel. Bei der Übersetzung wird immer die Regel mit der längeren Kette zuerst geprüft; Regeln mit gleichen Zeichen am Anfang werden also nach der Länge und bei gleicher Länge nach der alphabetischen Ordnung abgearbeitet.

Beispiel 12 enthält Ausschnitte aus GRWF und RSTA zu BODL.

# BEISPIEL 12.1

=====

|    |            |            |            |
|----|------------|------------|------------|
| 23 | TIR        | TYR        | T2R        |
| 2  | TIREN      | TYRE/      | T2R1N      |
| 1  | TIREN      | TYREN/     | T2R1N      |
| 1  | TIRES      | TYRES      | T2R1S      |
| 3  | TIRS       | TYRS       | T2RS       |
| 1  | FALSCH     | UAL\$CH    | F0L\$      |
| 2  | FALSCHEN   | UAL\$CHEN  | F0L\$1N    |
| 1  | FAS        | UA\$Z+     | F0S        |
| 1  | FAS        | UAS        | F0S        |
| 3  | UBER       | UBER       | 4B1R       |
| 2  | UBERWINDET | UBERWINDET | 4B1RW2ND1T |
| 1  | UBERWUND   | UBERWUND   | 4B1RW4ND   |
| 1  | UBERWUNDEN | UBERWUNDEN | 4B1RW4ND1N |
| 1  | FECHTENT   | UECHTENT   | F1CT1NT    |
| 1  | FELT       | UFLT       | F1LT       |
| 1  | FEST       | UE\$T      | F1ST       |
| 22 | FIL        | UIL        | F2L        |

# BEISPIEL 12.2

=====

| ORIGINAL |     | KONTEXT |    | ZAHL DER ANWENDUNGEN |        |
|----------|-----|---------|----|----------------------|--------|
| LESB.    |     | GRAPHEN |    | (VERSCH. WORTF.)     |        |
| GRAPHEN  |     |         |    |                      |        |
| UA       | FA  | 0       | F0 | 18                   | ( 12)  |
| UE       | FE  | 0       | F1 | 4                    | ( 4)   |
| UI       | FI  | 0       | F2 | 29                   | ( 3)   |
| UO       | FO  | 0       | F3 | 212                  | ( 36)  |
| U=       | U=  | 0       | 7  | 34                   | ( 30)  |
| U        | U   | 0       | 4  | 777                  | ( 267) |
| VA       | FA  | 0       | F0 | 16                   | ( 5)   |
| VEI      | FEI | 0       | F9 | 1                    | ( 1)   |
| VE       | FE  | 0       | F1 | 3                    | ( 2)   |
| VIE      | FI  | 0       | F2 | 3                    | ( 1)   |
| VI       | FI  | 0       | F2 | 36                   | ( 12)  |

### 3.1.14 Die Operation \*KGS[GRWF,RSTA,KWFP,VGRS;VGRS] =====

Der Operator \*KGS dient zum Iterieren des vorläufigen Graphemsystems an ein mögliches endgültiges. Durch Verwendung von RSTA können Regeln, die nicht benutzt werden, entfernt werden. Außerdem werden bei systematischen Fehlern ggf. Regeln entfernt, hinzugefügt oder ersetzt; dann ist der Funktionswert von \*KGS 'wahr'. Ist das endgültige Graphemsystem erreicht, wurde also durch \*KGS nichts mehr an VGRS geändert, so ist der Funktionswert 'falsch'.

Die Entscheidung ist hier natürlich auch weitgehend von dem verwendeten Graphembegriff abhängig; übrigens decken Differenzen zwischen Intention und Ausführung der Graphemschreibung (Übersetzung) häufig auch gewisse Unsauberkeiten bei dem zugrunde liegenden Graphembegriff auf. In vielen Fällen läßt sich auf dieser Stufe eine Klärung erreichen.

### 3.1.15 Die Operation \*EXR[RULE,RSTA;RRUL]

=====

Der Operator \*EXR ermöglicht eigentlich nur eine leichter lesbare Form der folgenden Ergebnisse. Zu jedem Einzelgraphem aus RULE wird in RRUL eine leicht lesbare Form angegeben. Jedes Element aus RRUL hat die Form:<Codierung des Graphems><lesbare Form des Graphems>. RRUL ließe sich auch leicht maschinell erstellen aus RULE; der manuelle Aufwand ist aber so gering, daß davon abgesehen wurde. Außerdem kann man in den meisten Fällen mit dem im Beispiel 13 gegebenen Standard-Vorschlag für RRUL (und für die Codierung in VGRS) auskommen.

BEISPIEL 13

=====

|    |     |
|----|-----|
| G  | A   |
| 1  | E   |
| 2  | I   |
| 3  | O   |
| 4  | U   |
| 5  | A=  |
| 6  | O=  |
| 7  | U=  |
| 8  | AU  |
| 9  | EI  |
| =  | EU  |
| ≠  | ≠   |
| ≤  | UE  |
| +  | UO  |
| B  | B   |
| C  | CH  |
| D  | D   |
| F  | F   |
| G  | G   |
| H  | H   |
| J  | J   |
| K  | K   |
| L  | L   |
| M  | M   |
| N  | N   |
| /  | NG  |
| P  | P   |
| )  | PF  |
| Q  | QU  |
| R  | R   |
| \$ | SCH |
| ↑  | SZ  |
| S  | S   |
| T  | T   |
| W  | W   |
| X  | X   |
| Z  | Z   |

### 3.1.16 Die Operation UEMG[GRWF,RRUL;UEGR,PUEG]

Der Operator UEMG erzeugt aus der Datei der Graphem-schreibungen GRWF unter Benutzung von RRUL die Matrix der Übergangshäufigkeiten von einem Graphem zum anderen; PUEG stellt den gedruckten Output dar, der absolute Häufigkeiten, relative Häufigkeiten der Grapheme und Übergänge bezüglich des 1. und bezüglich des 2. Zeichens (als 100%), diese Werte als Histogramme (mit log.Maßstab) und die zugehörigen Standardabweichungen sowie die Verteilung der Graphemzahl pro Wortform beinhaltet. UEGR enthält diese Ergebnisse (Output auf Lochkarte oder Platte) und dient als Input für den Operator KONX.

Ein Ausschnitt aus PUEG (Beispiel 14) soll hier zur Veranschaulichung dienen.

BEISPIEL 14.1  
=====

ZAHL ALLER GRAPHEME = 50314

| GRAPHEM | REL.HAEUF. | ABS.HAEUF. | STANDARDABW. |
|---------|------------|------------|--------------|
| E       | 9.9157     | 4989       | 0.1332       |
| I       | 6.0540     | 3046       | 0.1063       |
| U       | 3.9730     | 1999       | 0.0871       |
| A       | 3.1681     | 1594       | 0.0781       |
| O       | 2.3174     | 1166       | 0.0671       |
| EI      | 1.9557     | 984        | 0.0617       |
| AU      | 0.6599     | 332        | 0.0361       |
| EU      | 0.2425     | 122        | 0.0219       |
| U=      | 0.0696     | 35         | 0.0113       |
| O=      | 0.0398     | 20         | 0.0089       |
|         | 20.9246    | 10528      | 0.1813       |
| N       | 10.8161    | 5442       | 0.1385       |
| D       | 7.0756     | 3560       | 0.1143       |
| S       | 5.5174     | 2776       | 0.1018       |
| T       | 5.2570     | 2645       | 0.0995       |
| R       | 4.4163     | 2222       | 0.0916       |
| G       | 2.7269     | 1372       | 0.0726       |
| L       | 2.5162     | 1266       | 0.0698       |
| M       | 2.1366     | 1075       | 0.0645       |
| CH      | 1.8703     | 941        | 0.0604       |
| F       | 1.5523     | 781        | 0.0551       |
| W       | 1.3754     | 692        | 0.0519       |
| Z       | 1.2144     | 611        | 0.0488       |
| H       | 1.1786     | 593        | 0.0481       |
| B       | 0.9699     | 488        | 0.0437       |
| P       | 0.8944     | 450        | 0.0420       |
| K       | 0.7235     | 364        | 0.0378       |
| SCH     | 0.3319     | 167        | 0.0256       |
| PF      | 0.0835     | 42         | 0.0129       |
| J       | 0.0239     | 12         | 0.0069       |

BEISPIEL 14.2

=====

| LAENGE | LAUFENDE<br>WORTFORMEN | UNTERSCHIEDL.<br>WORTFORMEN |
|--------|------------------------|-----------------------------|
| 1      | 0                      | 0                           |
| 2      | 3                      | 1                           |
| 3      | 2194                   | 65                          |
| 4      | 4068                   | 258                         |
| 5      | 1267                   | 322                         |
| 6      | 1309                   | 427                         |
| 7      | 875                    | 394                         |
| 8      | 404                    | 231                         |
| 9      | 263                    | 139                         |
| 10     | 84                     | 72                          |
| 11     | 40                     | 32                          |
| 12     | 15                     | 11                          |
| 13     | 3                      | 3                           |
| 14     | 3                      | 2                           |

BEISPIEL 14.3  
=====

GRAPHEM E

ZAHLE DES 1.GRAPHEMS = 100%

| VOR | P       | SIGMA(P) |
|-----|---------|----------|
| E   | 0.2806  | 0.0749   |
| I   | 0.0000  | 0.0000   |
| U   | 0.0200  | 0.0200   |
| A   | 0.0000  | 0.0000   |
| O   | 0.0200  | 0.0200   |
| EI  | 0.0000  | 0.0000   |
| AU  | 0.0000  | 0.0000   |
| EU  | 0.0000  | 0.0000   |
| U=  | 0.0000  | 0.0000   |
| O=  | 0.0000  | 0.0000   |
|     | 7.4364  | 0.3714   |
| N   | 39.2063 | 0.6912   |
| D   | 0.4209  | 0.0917   |
| S   | 5.2916  | 0.3169   |
| T   | 4.4037  | 0.2907   |
| R   | 16.5163 | 0.5257   |
| G   | 0.8018  | 0.1263   |
| L   | 5.7527  | 0.3297   |
| M   | 6.7749  | 0.3558   |
| CH  | 3.1068  | 0.2456   |
| F   | 0.5612  | 0.1058   |
| W   | 1.4833  | 0.1711   |
| Z   | 3.4275  | 0.2576   |
| H   | 1.3029  | 0.1605   |
| B   | 1.5835  | 0.1767   |
| P   | 0.4009  | 0.0895   |
| K   | 0.6815  | 0.1165   |
| SCH | 0.5211  | 0.1019   |
| PF  | 0.0000  | 0.0000   |
| J   | 0.0000  | 0.0000   |

ZAHLE ALLER GRAPHEME = 100%

| P      | SIGMA(P) |
|--------|----------|
| 0.0273 | 0.0074   |
| 0.0000 | 0.0000   |
| 0.0020 | 0.0020   |
| 0.0000 | 0.0000   |
| 0.0020 | 0.0020   |
| 0.0000 | 0.0000   |
| 0.0000 | 0.0000   |
| 0.0000 | 0.0000   |
| 0.0000 | 0.0000   |
| 0.0000 | 0.0000   |
| 0.0000 | 0.0000   |
| 0.7374 | 0.0381   |
| 3.8876 | 0.0862   |
| 0.0417 | 0.0091   |
| 0.5247 | 0.0322   |
| 0.4373 | 0.0294   |
| 1.6377 | 0.0566   |
| 0.0795 | 0.0126   |
| 0.5704 | 0.0336   |
| 0.6718 | 0.0364   |
| 0.3081 | 0.0247   |
| 0.0557 | 0.0105   |
| 0.1471 | 0.0171   |
| 0.3399 | 0.0259   |
| 0.1292 | 0.0160   |
| 0.1570 | 0.0177   |
| 0.0398 | 0.0089   |
| 0.0676 | 0.0116   |
| 0.0517 | 0.0101   |
| 0.0000 | 0.0000   |
| 0.0000 | 0.0000   |

BEISPIEL 14.4

NACHBEREICH DES GRAPHEMS E (LOG.MASS.)

|     | -3 | -2    | -1 | 0 | 1 | 2 | -5 | -4    | -3 | -2 | -1 | 0 | 1 |
|-----|----|-------|----|---|---|---|----|-------|----|----|----|---|---|
|     | I  | I     | I  | I | I | I | I  | I     | I  | I  | I  | I | I |
| E   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| I   | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| U   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| A   | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| O   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| EI  | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| AU  | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| EU  | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| U=  | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| O=  | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
|     | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| N   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| D   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| S   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| T   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| R   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| G   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| L   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| M   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| CH  | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| F   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| W   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| Z   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| H   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| B   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| P   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| K   | I  | ***** |    |   |   |   | I  | ***** |    |    |    |   |   |
| SCH | I  | ***** |    |   |   |   | I  | ****  |    |    |    |   |   |
| PF  | I  |       |    |   |   |   | I  |       |    |    |    |   |   |
| J   | I  |       |    |   |   |   | I  |       |    |    |    |   |   |

### 3.1.17 Die Operation KONX[UEGR,METR;KONM,PKON]

=====

Der Operator KONX erstellt aus der Übergangsmatrix mit Nebenangaben UEGR und einer Abstandsdefinition<sup>1</sup> METR eine Zusammenhangsmatrix KONM. Die Definition einer Metrik METR kann etwa heißen: Benutze die relativen Häufigkeiten der Übergänge von oder/und zu einem Graphem als Ordinaten in einem  $n$ - (oder  $2n$ -) dimensionalen euklidischen Raum ( $n$ =Mächtigkeit der Menge der Grapheme) und berechne die euklidischen Abstände der Punkte von je zwei Graphemen. Daraus kann man ersehen, daß METR selbst ein Operator (Programmstück) sein muß. KONM ist eine  $n \times n$ -Matrix, in der die Abstände der Spalten- und Zeilen-Grapheme gespeichert werden (je nach aktueller Definition bezüglich des Vorbereichs, des Nachbereichs der Grapheme oder beider). Dieser Operator ist Teil eines Gesamtoperators, der außerdem CLSS enthält. PKON ist ein formatierter Output von KONM auf den Drucker (mit Zusatzinformation).

Beispiel 15 zeigt einen Ausschnitt aus PKON (BODL).

---

<sup>1</sup> Sehr allgemeine Ausführungen über die Problematik einer Metrik und Klassenbildung wurden in einem Kurzreferat vorgetragen: Straßner, E., Schönhut, J., Probleme einer Klassenbildung bei der graphematischen Analyse, gehalten am 11.6.1971 bei dem Symposium über Probleme der maschinellen Verarbeitung mittelhochdeutscher Texte beim Institut für deutsche Sprache, Mannheim.

BEISPIEL 15.1  
=====

| ZEICHEN<br>ABSTAND |     | NACHBEREICH | BEZUEGLICH<br>VORBEREICH | UMGEBUNG |
|--------------------|-----|-------------|--------------------------|----------|
| N                  | N   | 0.000       | 0.000                    | 0.000    |
| N                  | D   | 26.474      | 38.155                   | 46.440   |
| N                  | S   | 28.354      | 5.918                    | 28.965   |
| N                  | T   | 26.607      | 27.800                   | 38.481   |
| N                  | R   | 18.947      | 8.790                    | 20.886   |
| N                  | G   | 27.143      | 15.918                   | 31.467   |
| N                  | L   | 23.855      | 8.206                    | 25.227   |
| N                  | M   | 26.218      | 0.757                    | 26.229   |
| N                  | CH  | 29.411      | 4.340                    | 29.730   |
| N                  | F   | 25.934      | 2.328                    | 26.098   |
| N                  | W   | 26.519      | 5.269                    | 27.037   |
| N                  | Z   | 26.203      | 3.583                    | 26.447   |
| N                  | H   | 26.462      | 1.423                    | 26.500   |
| N                  | B   | 25.047      | 7.179                    | 26.056   |
| N                  | P   | 41.112      | 23.690                   | 47.449   |
| N                  | K   | 29.393      | 17.380                   | 34.147   |
| N                  | SCH | 35.340      | 16.900                   | 39.173   |
| N                  | PF  | 26.462      | 26.730                   | 37.613   |
| N                  | J   | 26.462      | 1.617                    | 26.511   |

BEISPIEL 15.2  
=====

RECHTE OBERE HAEFTE VORBEREICH  
LINKE UNTERE HAEFTE NACHBEREICH

|    | E   | I   | U   | A   | O   | EI  | AU  | EU  | U=  | O=  |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| E  | 0.0 | 0.3 | 0.2 | 0.3 | 0.2 | 0.3 | 0.6 | 0.3 | 0.3 | 0.3 |
| I  | 0.3 | 0.0 | 0.1 | 0.0 | 0.2 | 0.6 | 0.9 | 0.0 | 0.0 | 0.0 |
| U  | 0.3 | 0.1 | 0.0 | 0.1 | 0.2 | 0.5 | 0.8 | 0.1 | 0.1 | 0.1 |
| A  | 0.3 | 0.0 | 0.1 | 0.0 | 0.2 | 0.6 | 0.9 | 0.0 | 0.0 | 0.0 |
| O  | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 | 0.6 | 0.8 | 0.2 | 0.2 | 0.2 |
| EI | 0.3 | 0.0 | 0.1 | 0.0 | 0.1 | 0.0 | 0.3 | 0.6 | 0.6 | 0.6 |
| AU | 0.3 | 0.0 | 0.1 | 0.0 | 0.1 | 0.0 | 0.0 | 0.9 | 0.9 | 0.9 |
| EU | 0.3 | 0.0 | 0.1 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| U= | 0.3 | 0.0 | 0.1 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| O= | 0.3 | 0.0 | 0.1 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

### 3.1.18 Die Operation CLSS[KONM,SCHW;NTPL,VNTP]

=====

Der Operator CLSS erzeugt auf der Menge der Grapheme eine Überdeckung mit folgender Eigenschaft: Wenn der Abstand zweier Grapheme (Eintrag in KONM) kleiner oder gleich dem Schwellwert SCHW ist, sollen sie in Bezug auf ihre Umgebung (bzw. Vor- bzw. Nachbereich) ununterscheidbar<sup>1</sup> heißen. NTPL beinhaltet die Menge aller n-Tupel (jeweils nur das größte, nicht Teile von größeren), für die gilt: jedes Element des n-Tupels ist von jedem anderen des gleichen n-Tupels ununterscheidbar. VNTP gibt die Verteilung der n-Tupel in Form eines Histogramms. Eingabemedium für SCHW war zunächst Lochkarte oder Display<sup>2</sup>, da uns kein brauchbarer Algorithmus zur Berechnung von SCHW bekannt war. Ausgabemedium ist der Schnelldrucker (OUT) und zusätzlich bei Display-Input das Display, - dann gibt es zusätzlich Eingriffsmöglichkeiten in das laufende Programm (Abbruch, neues Aufsetzen an definierten Stellen, ändern gewisser Programmparameter) vom Bildschirm aus. Inzwischen ist dieses System jedoch mit einem Algorithmus zur Berechnung von SCHW ausgestattet, die den Input von SCHW über Lochkarte bzw. Display unnötig macht; damit entfällt auch der Output am Bildschirm.

Beispiel 16 enthält ein Teilprotokoll von NTPL und VNTP.

---

<sup>1</sup>Vgl. dazu:

Fischer, W.L., Toleranzräume, in: Archimedes, 22.Jg., Heft IV/1970, S.101-107, insbes.S.102.

<sup>2</sup>Es handelt sich hierbei um Datensichtgeräte mit einem Bildschirm und einer alphanumerischen Tastatur ähnlich einer Schreibmaschine.

BEISPIEL 16  
=====

TEXT BODL            SYSTEM DER GRAPHEME 2  
NACHBEREICH        SCHWELLWERT = 2.97

1-TUPEL

T01.01 = (N)  
T01.02 = (S)  
T01.03 = (R)  
T01.04 = (G)  
T01.05 = (L)  
T01.06 = (CH)  
T01.07 = (F)  
T01.08 = (Z)  
T01.09 = (B)  
T01.10 = (P)  
T01.11 = (K)  
T01.12 = (SCH)

2-TUPEL

T02.01 = (D,T)

4-TUPEL

T04.01 = (M,H,PF,J)

5-TUPEL

T05.01 = (D,W,H,PF,J)

VERTEILUNG DER N-TUPEL BEI SCHWELLWERT 2.97

|         |    |        |
|---------|----|--------|
| 1-TUPEL | 12 | I***** |
| 2-TUPEL | 1  | I*     |
| 3-TUPEL | 0  | I      |
| 4-TUPEL | 1  | I*     |
| 5-TUPEL | 1  | I*     |

### 3.1.18.1 Exkurs 9: Algorithmus zur Berechnung der Schwellwerte

Wichtig für die Festlegung des Schwellwertes SCHW war unser Interesse, die Veränderung der Struktur des Systems bei Veränderung von SCHW zu beobachten. Es gibt dabei zwei Extremwerte die nicht unter- bzw. überschritten werden sollten: Das Minimum liegt dann vor, wenn SCHW gerade klein genug ist, daß jedes Element nur von sich selbst ununterscheidbar ist, das Maximum, wenn SCHW gerade so groß ist, daß alle Elemente voneinander ununterscheidbar sind. Im ersten Fall bekommen wir  $n$  Toleranzklassen<sup>1</sup> mit je einem Graphem, im zweiten eine Toleranzklasse mit genau  $n$  Graphemen als Elementen. In beiden Fällen gilt auch das transitive Gesetz; daher sind sie Äquivalenzklassen.

Die dazwischenliegenden feineren und groberen Überdeckungen des Graphemsystems sind die uns interessierenden Strukturen. Erhöht man den Schwellwert ausgehend von Minimum immer auf den nächstgrößeren Wert aus der Abstandsmatrix bis zum Maximum, so erhält man alle möglichen Zwischenstufen der Überdeckungen. Für die Praxis wird es genügen, hier eine beschränkte Zahl anzugeben, um den Rechenzeitbedarf nicht unnötig zu vergrößern.

---

<sup>1</sup> Zu dem Problem der Toleranz- bzw. Äquivalenzrelation und der durch sie implizierten Klassenbildung vgl. die Arbeit: Fischer, W.L., Äquivalenz- und Toleranzstrukturen in der Linguistik - Zur Theorie der Synonyma, im Druck.  
Für die Überlassung eines Manuskriptes dieser Arbeit sowie für seine unermüdliche Bereitschaft zur Diskussion dieser Probleme möchte ich Herrn Dr. Fischer an dieser Stelle herzlich danken.

3.1.19 Die Operation \*KCS[NTPL,GRWF,RSTA,KWFP,KREG,VGRS;VGRS]  
=====

Der Operator \*KCS ermöglicht die Korrektur des Graphemsystems, wenn sich bei der Überdeckung der Grapheme etwas fehlerhaftes erwiesen haben sollte. Wenn man etwa die Zeichenfolge NG in dieser Verbindung als Graphem /NG/, sonst als /N/ und /G/ ins Graphemsystem aufgenommen hat, so zeigt sich möglicherweise, daß /NG/ im Vorbereich ununterscheidbar von /N/, im Nachbereich ununterscheidbar von /G/ ist. Liegt solch ein Fall vor, so ist das Graphem /NG/ aus dem System zu entfernen.

Bei Veränderung von VGRS durch \*KCS hat die Operation den Funktionswert 'wahr', sonst 'falsch'.

### 3.1.2o Die Operation WSEG [GRWF,RRUL;SEGW]

=====

Der Operator WSEG besteht im wesentlichen aus einem Segmentier-Algorithmus für die Wortformen in GRWF; RRUL ermöglicht es, die Ergebnisse der Segmentierung SEGW in leicht lesbarer Form auszugeben. Der Segmentier-Algorithmus geht davon aus, daß in jeder Wortform mindestens ein Graphem des Typs 1, nicht notwendig aber ein Graphem des Typs 2, enthalten sein muß. Ein Segment (oder Wortkonstituent) besteht daher aus einem Graphem vom Typ 1; davor (2-Radikal) und/oder danach (1-Radikal) können noch ein oder mehrere zum Segment gehörende Grapheme vom Typ 2 stehen. Je nach Besetzung des 1- und 2-Radikals eines Wortkonstituenten unterscheiden wir 4 Typen:

Typ 0: 2-Radikal unbesetzt, 1-Radikal unbesetzt

Typ 1: 2-Radikal unbesetzt, 1-Radikal besetzt

Typ 2: 2-Radikal besetzt, 1-Radikal unbesetzt

Typ 3: 2-Radikal besetzt, 1-Radikal besetzt.

Die Grenze zwischen zwei Wortkonstituenten ist abhängig vom Abstand (Zahl der Grapheme von Typ 2) der zugehörigen Grapheme von Typ 1.

| Abstand d | Grenze nach n Graphemen<br>nach dem 1. Graphem vom Typ 1 |
|-----------|----------------------------------------------------------|
| d=0       | n=0                                                      |
| 0<d<2     | n=1                                                      |
| d<2       | n=2                                                      |

Liegt allerdings ein Segmentgrennzeichen (.) vor, so wird unabhängig vom Segmentieralgorithmus an dieser Stelle segmentiert. Solche Segmentgrennzeichen können durch die "Praefix"-Abspaltung (dort eingefügter \*) oder durch entsprechende Regeln in VGRS bzw. RULE entstanden sein.

Unter Wortkonstituentenstruktur WKS verstehen wir dann die lineare Folge der in der Wortform vorkommenden Wortkonstituententypen; so hat z.B.

WA/REN die WKS 2 3.

Die Ausgabe SEGW gliedert sich in 9 Unterdateien:

1. Die Aufspaltung der Wortform in Wortkonstituenten mit Häufigkeitsangabe.
2. Wortkonstituenten alphabetisch sortiert, Ausgabe initialer oder finaler Position in der Wortform, zusätzlich Wortform in Original- und Graphembeschreibung.
3. Wortkonstituenten, aufgespalten in 2-Radikal, Kern, 1-Radikal, alphabetisch sortiert, mit Häufigkeitsangabe, Wortkonstituententyp und Angabe bei Final- oder Initialposition.
4. 2-Radikale, sortiert mit Angabe der Graphembeschreibung der Wortform, des Wortkonstituenten und des Wortkonstituententyps.
5. Wie 4. für Kerne.
6. Wie 4. für 1-Radikale.
7. Übergänge von Wortkonstituenten, sortiert mit Graphembeschreibung der Wortform und Häufigkeitsangabe.
8. Graphembeschreibung, sortiert mit Originalschreibung und Häufigkeit.
9. WKS, Segmentzahl, Normalschreibung der Wortform und Häufigkeit.

Diese Dateien liegen auch in Form von Listen vor. Auszüge aus diesen Listen mögen zur Veranschaulichung dienen. (Beispiel 17)

BEISPIEL 17.1

=====

|    |      |       |       |      |
|----|------|-------|-------|------|
| 3  | GA   | BEN   |       |      |
| 1  | GANZ |       |       |      |
| 1  | GANG |       |       |      |
| 1  | GAR  |       |       |      |
| 1  | GAR  | TEN   |       |      |
| 2  | GEB  |       |       |      |
| 6  | GE   | BEN   |       |      |
| 14 | GE   | BEN   |       |      |
| 1  | GE   | BEN   |       |      |
| 1  | GE   | BENS  |       |      |
| 1  | GE   | BENT  |       |      |
| 3  | GEBT |       |       |      |
| 4  | GE   | KLEI  | DET   |      |
| 2  | GE   | KLEIT |       |      |
| 1  | GE   | KLEIT |       |      |
| 4  | GE   | KLEIT |       |      |
| 1  | GE   | KREU  | ZIGT  |      |
| 1  | GE   | KREU  | ZIG   | TE   |
| 1  | GE   | ZEI   | CHENT |      |
| 12 | GE   | ZEI   | CHENT |      |
| 1  | GE   | ZEI   | CHEN  | TEN  |
| 1  | GE   | ZE    | LEN   |      |
| 3  | GE   | ZEU   | GE    |      |
| 1  | GE   | ZEU   | GEN   |      |
| 1  | GE   | ZEU   | GINS  |      |
| 7  | GE   | ZEUG  | NIS   |      |
| 1  | GE   | ZEUG  | NIS   |      |
| 1  | GE   | ZEUG  | NIS   |      |
| 1  | GE   | ZEUG  | NU=S  |      |
| 1  | GE   | ZEUG  | NIS   |      |
| 2  | GE   | ZIRT  |       |      |
| 1  | GE   | ZUKT  |       |      |
| 2  | GE   | DECH  | TIG   |      |
| 1  | GE   | DECHT | NIS   |      |
| 1  | GE   | DUL   | DET   |      |
| 4  | GE   | DULT  |       |      |
| 2  | GE   | DUL   | TI    | KEIT |
| 1  | GE   | FUGT  |       |      |

# BEISPIEL 17.2

=====

|   |       |   |               |               |
|---|-------|---|---------------|---------------|
| 2 | LEN   | * | SELEN         | \$ELEN        |
| 1 | LEN   | * | SOLEN         | \$OLLEN       |
| 2 | LEN   | * | STOLEN        | \$TOLE/       |
| 1 | LEN   | * | STOLEN        | \$TOLEN       |
| 1 | LEN   | * | STOLEN        | \$TOLEN/      |
| 1 | LEN   | * | STOLEN        | \$TOLLE/      |
| 1 | LEN   | * | STU=LEN       | \$TU=LE/      |
| 3 | LEN   | * | SULEN         | \$ULLE/       |
| 2 | LEN   | * | SULEN         | \$ULLEN       |
| 3 | LEN   | * | TEILEN        | TEILEN        |
| 1 | LEN   | * | URTEILEN      | URTEILEN      |
| 1 | LEN   | * | WILEN         | WILLE/        |
| 2 | LEN   | * | WOLEN         | WOLLE/        |
| 1 | LER   | * | AD.LER        | ADLER         |
| 3 | LER   | * | ALER          | ALLER         |
| 7 | LES   | * | ALES          | ALLES         |
| 3 | LET   | * | FULET         | FULLET        |
| 1 | LET   | * | WALET         | WALLET        |
| 1 | LEU   |   | FORLEUSE      | VORLEU\$E     |
| 1 | LEU   |   | FORLEUTEN     | UORLEUTEN     |
| 1 | LEU   |   | URLEUGE       | VRLEUGE       |
| 1 | LEU   |   | URLEUGE       | VRLEWGE       |
| 1 | LEUCH |   | DURCHLEUCHTIG | DURCHLEUCHTIG |
| 1 | LEUG  | * | URLEUG        | VRLEU=G       |
| 3 | LEUG  | * | URLEUG        | VRLEUG        |
| 1 | LEUG  | * | URLEUG        | VRLEWG        |
| 3 | LEUT  | * | KAUFLEUT      | KAUFFLEUT     |
| 1 | LEUT  | * | SCHIFLEUT     | \$CHIFFLEWT   |
| 1 | LI    |   | FORLISEN      | UORLI\$EN     |

BEISPIEL 17.3

=====

|   |    |    |     |   |   |                 |
|---|----|----|-----|---|---|-----------------|
| 3 | D  | EI | N   | * | 3 | GULDEIN         |
| 1 | GR | EI | F   | * | 3 | BE.GREIF        |
| 1 | H  | EI | T   | * | 3 | GESUNT.HEIT     |
| 1 | H  | EI | T   | * | 3 | GE.FANGENHEIT   |
| 1 | H  | EI | T   | * | 3 | GE.FANGENHEIT   |
| 1 | H  | EI | T   | * | 3 | GOT.HEIT        |
| 3 | H  | EI | T   | * | 3 | KLARHEIT        |
| 1 | H  | EI | T   | * | 3 | PLOSHEIT        |
| 1 | H  | EI | T   | * | 3 | POSHEIT         |
| 1 | H  | EI | T   | * | 3 | STARKHEIT       |
| 4 | H  | EI | T   | * | 3 | WEISHEIT        |
| 1 | K  | EI | T   | * | 3 | DEMU=TIKEIT     |
| 1 | K  | EI | T   | * | 3 | FORMEINSAMIKEIT |
| 2 | K  | EI | T   | * | 3 | GEDULTIKEIT     |
| 2 | K  | EI | T   | * | 3 | GERECHTIKEIT    |
| 1 | K  | EI | T   | * | 3 | GLUSTIKEIT      |
| 1 | K  | EI | T   | * | 3 | HEIMLIKEIT      |
| 1 | K  | EI | T   | * | 3 | SCHAMIKEIT      |
| 1 | K  | EI | T   | * | 3 | UNREINIKEIT     |
| 1 | K  | EI | T   | * | 3 | UNWIRDIKEIT     |
| 1 | K  | EI | T   | * | 3 | ZWIFALTIKEIT    |
| 1 | KL | EI | T   | * | 3 | GE.KLEIT        |
| 2 | KL | EI | T   | * | 3 | GE.KLEIT        |
| 4 | KL | EI | T   | * | 3 | GE.KLEIT        |
| 2 | KR | EI | S   | * | 3 | UMB.KREIS       |
| 1 | KR | EI | S   | * | 3 | UM.KREIS        |
| 1 | L  | EI | CH  | * | 3 | ENLEICH         |
| 2 | L  | EI | N   | * | 3 | ALEIN           |
| 1 | L  | EI | N   | * | 3 | TUCHELEIN       |
| 1 | M  | EI | N   | * | 3 | FORMEIN         |
| 1 | M  | EI | N   | * | 3 | GEMEIN          |
| 6 | N  | EI | N   | * | 3 | EIS.NEIN        |
| 1 | N  | EI | N   | * | 3 | STEINEIN        |
| 1 | R  | EI | CHS | * | 3 | ERTREICHS       |
| 1 | R  | EI | CHT | * | 3 | GEREICHT        |
| 2 | R  | EI | N   | * | 3 | EREIN           |
| 1 | R  | EI | N   | * | 3 | HEREIN          |
| 1 | R  | EI | NS  | * | 3 | UNREINS         |
| 1 | R  | EI | T   | * | 3 | BEREIT          |
| 3 | R  | EI | T   | * | 3 | BEREIT          |

BEISPIEL 17.4

=====

|    |   |        |   |                 |
|----|---|--------|---|-----------------|
| 1  | M | MEN    | * | GENUMEN         |
| 6  | M | MEN    | * | KUMEN           |
| 1  | M | MEN    | * | KUMEN           |
| 22 | M | MEN    | * | NAMEN           |
| 1  | M | MEN    | * | NAMEN           |
| 5  | M | MEN    | * | NAMEN           |
| 1  | M | MEN    | * | PAUMEN          |
| 2  | M | MEN    | * | SAMEN           |
| 1  | M | MEN    | * | SAMEN           |
| 2  | M | MEN    | * | SAMEN           |
| 1  | M | MEN    | * | SCHAMEN         |
| 1  | M | MEN    | * | STIMEN          |
| 1  | M | MEND   | * | KLIMEND         |
| 1  | M | MENS   | * | NAMENS          |
| 1  | M | MENS   | * | NAMENS          |
| 1  | M | MENT   | * | FUNDAMENT       |
| 1  | M | MENT   | * | FUNDAMENT       |
| 1  | M | MENT   | * | KLIMENT         |
| 1  | M | MENT   | * | KU=MENT         |
| 1  | M | MENT   | * | KUMENT          |
| 1  | M | MENT   | * | TORMENT         |
| 1  | M | MER    | * | SUMER           |
| 1  | M | MER    | * | TEILNEMER       |
| 2  | M | MES    | * | LAMES           |
| 6  | M | MES    | * | LAMES           |
| 1  | M | MET    | * | AB.NIMET        |
| 2  | M | MET    | * | GESAMET         |
| 3  | M | MISCHT | * | GEMISCHT        |
| 1  | M | MUNG   | * | FORMEINSAMUNG   |
| 1  | M | MUT    | * | ARMUT           |
| 2  | M | MUT    | * | WERMUT          |
| 1  | M | MA     |   | RECHTMACHUNG    |
| 1  | M | MAN    |   | KAUFMANSCHAFT   |
| 2  | M | ME     |   | GEMESEN         |
| 1  | M | ME     |   | GEMESEN         |
| 3  | M | MECH   |   | ALMECHTIG       |
| 2  | M | MECH   |   | ALMECHTIGEN     |
| 2  | M | MECH   |   | ALMECHTIGER     |
| 2  | M | MEI    |   | GEMEINEN        |
| 1  | M | MEIN   |   | FORMEINSAMIKEIT |
| 1  | M | MEIN   |   | FORMEINSAMUNG   |
| 1  | M | MI     |   | FORMEINSAMIKEIT |
| 1  | M | MI     |   | GEMISCHET       |
| 1  | M | MI     |   | SCHAMIKEIT      |
| 1  | M | MI     |   | UNSCHAMIGEN     |

# BEISPIEL 17.5

=====

|    |   |   |       |                  |
|----|---|---|-------|------------------|
| 1  | U | * | UR    | URKUND           |
| 1  | U | * | UR    | URKUNDES         |
| 3  | U | * | UR    | URLEUG           |
| 1  | U | * | UR    | URLEUG           |
| 1  | U | * | UR    | URLEUG           |
| 1  | U | * | UR    | URLEUGE          |
| 1  | U | * | UR    | URLEUGE          |
| 5  | U | * | UR    | URTEIL           |
| 1  | U | * | UR    | URTEILEN         |
| 1  | U | * | UR    | URTEILS          |
| 1  | U | * | UR    | URTEILST         |
| 1  | U | * | UR    | URTEILT          |
| 1  | U | * | UR    | URTEILTE         |
| 1  | U | * | WUN   | WUNDEN           |
| 1  | U | * | WUN   | WUNDER           |
| 2  | U | * | WUN   | WUNDERLICH       |
| 1  | U | * | WUN   | WUNDERNT         |
| 1  | U | * | WUN   | WUNDERT          |
| 1  | U | * | WUN   | WUNDERUNG        |
| 1  | U | * | WUR   | WURDE            |
| 2  | U | * | WUR   | WURDEN           |
| 1  | U | * | WUR   | WURDEN           |
| 2  | U | * | WUR   | WURZEL           |
| 1  | U | * | ZU    | ZUGEN            |
| 1  | U | * | ZU    | ZUHANT           |
| 2  | U | * | ZU    | ZUHANT           |
| 1  | U | * | ZU    | ZUSEZT           |
| 1  | U | * | ZUN   | ZUNOEN           |
| 4  | U | * | ZUN   | ZUNGEN           |
| 2  | U | * | ZUN   | ZUNGEN           |
| 1  | U | * | ZUN   | ZUNGEN           |
| 1  | U | * | ZUN   | ZUNGEN           |
| 3  | U |   | CHUNG | * BEZEICHUNG     |
| 1  | U |   | CHUNG | * RECHTMACHUNG   |
| 3  | U |   | CHUNG | * SPRECHUNG      |
| 1  | U |   | CHUNG | * WAL. SPRECHUNG |
| 4  | U |   | DULT  | * GEDULT         |
| 1  | U |   | DUNG  | * ENZUNGUNG      |
| 2  | U |   | DUNG  | * STENDUNG       |
| 1  | U |   | DURST | * ENDURST        |
| 1  | U |   | FUGT  | * GEFUGT         |
| 2  | U |   | GRUND | * AB. GRUND      |
| 1  | U |   | GRUNT | * AB. GRUNT      |
| 1  | U |   | GUNG  | * BE. WEGUNG     |
| 2  | U |   | GURT  | * BEGURT         |
| 2  | U |   | HUT   | * BEHUT          |
| 1  | U |   | KUND  | * URKUND         |
| 1  | U |   | KUNG  | * DANKUNG        |
| 1  | U |   | LUNG  | * TRUBSALUNG     |
| 4  | U |   | LUNG  | * TRUBSALUNG     |
| 11 | U |   | LUZ   | * ANT. LUZ       |
| 1  | U |   | MUNG  | * FORMEINSAMUNG  |
| 1  | U |   | MUT   | * ARMUT          |
| 2  | U |   | MUT   | * WERMUT         |

BEISPIEL 17.6

=====

|     |     |   |        |   |           |
|-----|-----|---|--------|---|-----------|
| 1   | BST | * | LEBST  | * | LEBST     |
| 3   | BT  | * | GEBT   | * | GEBT      |
| 2   | BT  | * | GIBT   | * | GIBT      |
| 2   | BT  | * | HABT   | * | HABT      |
| 13  | BT  | * | HAUBT  | * | HAUBT     |
| 1   | BT  | * | HEBT   | * | HEBT      |
| 1   | BT  | * | LEBT   | * | LEBT      |
| 1   | BT  | * | HABT   | * | GEHABT    |
| 1   | BT  | * | LEBT   | * | GELEBT    |
| 7   | CH  | * | AUCH   | * | AUCH      |
| 8   | CH  | * | DICH   | * | DICH      |
| 1   | CH  | * | DOCH   | * | DOCH      |
| 10  | CH  | * | EUCH   | * | EUCH      |
| 23  | CH  | * | GLEICH | * | GLEICH    |
| 2   | CH  | * | HOCH   | * | HOCH      |
| 148 | CH  | * | ICH    | * | ICH       |
| 22  | CH  | * | ICH    | * | ICH       |
| 3   | CH  | * | MACH   | * | MACH      |
| 5   | CH  | * | MICH   | * | MICH      |
| 5   | CH  | * | NACH   | * | NACH      |
| 67  | CH  | * | NOCH   | * | NOCH      |
| 2   | CH  | * | PAUCH  | * | PAUCH     |
| 26  | CH  | * | PUCH   | * | PUCH      |
| 12  | CH  | * | RAUCH  | * | RAUCH     |
| 14  | CH  | * | REICH  | * | REICH     |
| 1   | CH  | * | SACH   | * | SACH      |
| 1   | CH  | * | SACH   | * | SACH      |
| 43  | CH  | * | SACH   | * | SACH      |
| 6   | CH  | * | SICH   | * | SICH      |
| 1   | CH  | * | SICH   | * | SICH      |
| 23  | CH  | * | SICH   | * | SICH      |
| 16  | CH  | * | SPRACH | * | SPRACH    |
| 1   | CH  | * | SPRECH | * | SPRECH    |
| 1   | CH  | * | SPRICH | * | SPRICH    |
| 1   | CH  | * | SUCH   | * | SUCH      |
| 8   | CH  | * | TRACH  | * | TRACH     |
| 2   | CH  | * | TUCH   | * | TUCH      |
| 1   | CH  | * | FECH   |   | FECHTENT  |
| 2   | CH  | * | GLEICH |   | GLEICHNIS |
| 1   | CH  | * | GLEICH |   | GLEICHNUS |

BEISPIEL 17.7

=====

|   |     |         |              |
|---|-----|---------|--------------|
| 2 | BE  | DARF    | BEDARF       |
| 1 | BE  | DEKT    | BEDEKT       |
| 1 | BE  | DUR     | BEDURFEN     |
| 3 | BE  | FLEKT   | BE.FLEKT     |
| 1 | BE  | GE      | BEGERENT     |
| 1 | BE  | GE      | BEGERUNG     |
| 1 | BE  | GI      | BEGINET      |
| 1 | BE  | GIN     | BEGIN        |
| 1 | BE  | GREIF   | BE.GREIF     |
| 1 | BE  | GRI     | BE.GRIFEN    |
| 2 | BE  | GURT    | BEGURT       |
| 3 | BE  | HAL     | BEHALTEN     |
| 1 | BE  | HAL     | BEHALTENT    |
| 2 | BE  | HALT    | BEHALT       |
| 1 | BE  | HEG     | BEHEG.LICH   |
| 1 | BE  | HEL     | BEHELTET     |
| 2 | BE  | HU      | BEHUTEN      |
| 1 | BE  | HU=T    | BEHU=T       |
| 2 | BE  | HUT     | BEHUT        |
| 1 | BE  | KERT    | BEKERT       |
| 1 | BE  | KO      | BEKORUNG     |
| 2 | BE  | REI     | BEREITET     |
| 1 | BE  | REI     | FORBEREITEN  |
| 3 | BE  | REIT    | BEREIT       |
| 3 | BE  | REIT    | BEREIT       |
| 1 | BE  | SE      | FORBESEZUNG  |
| 1 | BE  | SPRENGT | BE.SPRENGT   |
| 1 | BE  | STAT    | BE.STAT      |
| 1 | BE  | STEN    | BE.STEN      |
| 1 | BE  | SUND    | BESUNDERLICH |
| 1 | BE  | TRACHT  | BE.TRACHT    |
| 1 | BE  | WE      | BE.WEGET     |
| 1 | BE  | WE      | BE.WEGUNG    |
| 1 | BE  | WEG     | BE.WEG       |
| 1 | BE  | WEGT    | BE.WEGT      |
| 2 | BE  | ZEI     | BEZEICHEN    |
| 1 | BE  | ZEI     | BEZEICHENST  |
| 2 | BE  | ZEI     | BEZEICHENT   |
| 1 | BE  | ZEI     | BEZEICHET    |
| 3 | BE  | ZEI     | BEZEICHUNG   |
| 1 | BE  | ZEU     | BEZEUGEN     |
| 1 | BE  | ZEU     | BEZEUGER     |
| 1 | BE  | SLOS    | BE.SLOS      |
| 1 | BEI | TEN     | ARBEITEN     |
| 1 | BEI | TEN     | ARBEITEN     |

BEISPIEL 17.3  
=====

|              |                 |    |
|--------------|-----------------|----|
| ARMEN        | ARMEN           | 1  |
| ARMUT        | ARMUT           | 1  |
| AS           | A\$Z+           | 1  |
| AUCH         | AUCH            | 7  |
| AUF          | AUFF            | 75 |
| AUF          | AUF             | 34 |
| AUFGANG      | AUFFGANG        | 1  |
| AUGE         | AUGE            | 1  |
| AUGEN        | AUGEN/          | 1  |
| AUGEN        | AUGEN           | 3  |
| AUGEN        | AUGE/           | 6  |
| AUS          | AU\$            | 1  |
| AUS          | AUS             | 65 |
| AUS          | AU\$Z+          | 1  |
| AUSEN        | AU\$\$EN        | 2  |
| AUSERHALBEN  | AU\$\$ERTHALBE/ | 1  |
| AUSWENDIG    | AUSWENOIG       | 1  |
| BANG         | BANG            | 1  |
| BEDARF       | BEDARFF         | 2  |
| BEDEKT       | BEDEKT          | 1  |
| BEDURFEN     | BEDURFFEN       | 1  |
| BEGERENT     | BEGERENT        | 1  |
| BEGERUNG     | BEGERUN/G       | 1  |
| BEGIN        | BEGYN/          | 1  |
| BEGINET      | BEGINNET        | 1  |
| BEGURT       | BEGURT          | 2  |
| BEHALT       | BEHALT          | 2  |
| BEHALTEN     | BEHALTE/        | 3  |
| BEHALTENT    | BEHALTENT       | 1  |
| BEHEG.LICH   | BEHEGLICH       | 1  |
| BEHELTET     | BEHELTET        | 1  |
| BEHU=T       | BEHU=T          | 1  |
| BEHUT        | BEHUTT          | 2  |
| BEHUTEN      | BEHUTEN         | 2  |
| BEKERT       | BEKERT          | 1  |
| BEKORUNG     | BEKORUNG        | 1  |
| BEREIT       | BEREITT         | 3  |
| BEREIT       | BEREIT          | 1  |
| BEREITET     | BEREITET        | 2  |
| BESUNDERLICH | BE\$UNDERLICH   | 1  |
| BEZEICHEN    | BECZ+EICHEN     | 2  |
| BEZEICHENST  | BECZ+EICHEN\$T  | 1  |
| BEZEICHENT   | BECZ+EICHENT    | 2  |
| BEZEICHET    | BECZ+EICHET     | 1  |
| BEZEICHUNG   | BECZ+EICHUNG    | 3  |
| BEZEUGEN     | BECZ+EUGE/      | 1  |

BEISPIEL 17.9

=====

|   |   |                    |   |   |       |
|---|---|--------------------|---|---|-------|
| 1 | 3 | TEILHAFTIG         | 3 | 3 | 3     |
| 1 | 3 | UORDURBNISZ+       | 3 | 3 | 3     |
| 1 | 3 | UORGANGE/          | 3 | 3 | 3     |
| 1 | 3 | UORGANGEN/         | 3 | 3 | 3     |
| 2 | 3 | UORGULDET          | 3 | 3 | 3     |
| 1 | 3 | UORPORGE/          | 3 | 3 | 3     |
| 1 | 3 | UORPORGEN          | 3 | 3 | 3     |
| 1 | 3 | VINSTERNISZ+       | 3 | 3 | 3     |
| 1 | 3 | VORDAMPNISZ+       | 3 | 3 | 3     |
| 1 | 3 | VORDURPNUSZ+       | 3 | 3 | 3     |
| 1 | 3 | VORGANGE/          | 3 | 3 | 3     |
| 1 | 3 | VORGISTEN          | 3 | 3 | 3     |
| 1 | 3 | VORHENGEN          | 3 | 3 | 3     |
| 1 | 3 | VORWUSTET          | 3 | 3 | 3     |
| 1 | 3 | WEIN*GARTE/        | 3 | 3 | 3     |
| 1 | 3 | WEIN*TRAUBTE/      | 3 | 3 | 3     |
| 2 | 3 | WUNDERLICH         | 3 | 3 | 3     |
| 1 | 4 | OFFENBARUNG        | 0 | 3 | 2 3   |
| 1 | 4 | OFFENLICHEN        | 0 | 3 | 2 3   |
| 1 | 4 | AUSSERTHALBE/      | 0 | 3 | 3 3   |
| 2 | 4 | UBERWINDET         | 0 | 3 | 3 3   |
| 1 | 4 | UBERWUNDEN         | 0 | 3 | 3 3   |
| 1 | 4 | ANZ+UPETE/         | 1 | 2 | 2 3   |
| 4 | 4 | IN*GESIGEL         | 1 | 2 | 2 3   |
| 1 | 4 | IN*GESIGELN        | 1 | 2 | 2 3   |
| 1 | 4 | VN*GLAUBIGEN       | 1 | 2 | 2 3   |
| 1 | 4 | VNREINIKEIT        | 1 | 2 | 2 3   |
| 1 | 4 | VN\$CHAMIGE/       | 1 | 2 | 2 3   |
| 1 | 4 | VN*\$TU=MIGLICH    | 1 | 2 | 3 3   |
| 1 | 4 | AB*GE\$NITE/       | 1 | 3 | 2 3   |
| 2 | 4 | ALMECHTIGE/        | 1 | 3 | 2 3   |
| 2 | 4 | ALMECHTIGER        | 1 | 3 | 2 3   |
| 1 | 4 | VNWIRDIKEIT        | 1 | 3 | 2 3   |
| 1 | 4 | GE*VNKEUSCHET      | 2 | 1 | 2 3   |
| 1 | 4 | DEMU=TIKEIT        | 2 | 2 | 2 3   |
| 1 | 4 | GE*CREUCZ+IGTE     | 2 | 2 | 3 2   |
| 1 | 4 | GECZ+EICHENTEN/    | 2 | 2 | 3 3   |
| 2 | 4 | GEDULTIKEIT        | 2 | 3 | 2 3   |
| 2 | 4 | GERECHTIKEIT       | 2 | 3 | 2 3   |
| 1 | 4 | LEBENTIGEN/        | 2 | 3 | 2 3   |
| 1 | 4 | Z+HYFALTIKEIT      | 2 | 3 | 2 3   |
| 1 | 4 | BESUNDERLICH       | 2 | 3 | 3 3   |
| 1 | 4 | GE*UANGENHAIT      | 2 | 3 | 3 3   |
| 1 | 4 | GE*UANGENHEIT      | 2 | 3 | 3 3   |
| 1 | 4 | Z+HIFALTICLICH     | 2 | 3 | 3 3   |
| 1 | 4 | VIR*VNDVIRCZ+IK    | 3 | 1 | 3 3   |
| 2 | 4 | VIR*VNDZ+WEINCZ+IG | 3 | 1 | 3 3   |
| 1 | 4 | UORBE\$ECZ+UNG     | 3 | 2 | 2 3   |
| 1 | 4 | VORBEREITTE/       | 3 | 2 | 2 3   |
| 3 | 4 | \$UNDERLICHEN      | 3 | 3 | 2 3   |
| 1 | 4 | UORMEIN\$AMU/G     | 3 | 3 | 2 3   |
| 1 | 5 | VORMAIN\$AMIKEIT   | 3 | 3 | 2 2 3 |

3.1.21 Die Operation WKST[SEGW;WKH,WKSH]  
=====

Der Operator WKST benutzt die 9. Unterdatei von SEGW und erstellt daraus eine Statistik der Wortkonstituenten WKH und der Wortkonstituentenstrukturen WKSH. Es gibt zwei Versionen von WKST, wobei in der zweiten zu jeder Wortkonstituentenstruktur und deren Häufigkeit noch sämtliche Belegformen dieser Struktur ausgegeben werden. Dies erleichtert die Belegsuche sehr, so daß auch der Vergleich der WKSH verschiedener Texte gut möglich ist.

In Beispiel 18 sind Ausschnitte aus WKH und WKSH des Textes BODL angeführt.

BEISPIEL 18.1

=====

TEXTKENNWERTE

\*\*\*\*\*

TEXT BODL BODLEIANA

ZAHL DER WOERTER W = 10528  
ZAHL DER SEGMENTE S = 14305  
ZAHL DER WORTFORMEN F = 1357

VERTEILUNG DER SEGMENTZAHL/WORT L(W)

-----

| L(W) | BELEGHAEUFIGKEIT P(L(W)) |
|------|--------------------------|
|      | ABSOLUT RELATIV          |
| 1    | 7365 69.9563             |
| 2    | 2596 24.6581             |
| 3    | 521 4.9487               |
| 4    | 45 0.4274                |
| 5    | 1 0.0095                 |
| 6    | 0 0.0000                 |

HAEUFIGKEITSVERTEILUNG DER SEGMENTTYPEN

-----

| SEGMENTTYP | BELEGHAEUFIGKEIT P(S) |
|------------|-----------------------|
|            | ABSOLUT RELATIV       |
| 0          | 265 1.8455            |
| 1          | 3020 21.1115          |
| 2          | 3766 26.3265          |
| 3          | 7255 50.7165          |

BEISPIEL 18.2

=====

VERTEILUNG DER WORTKONSTITUENTENSTRUKTUREN

| STRUKTUR  | ZAHL | RELATIV |
|-----------|------|---------|
| 0         | 3    | 0.0285  |
| 1         | 2558 | 24.2971 |
| 2         | 1033 | 10.3343 |
| 3         | 3716 | 35.2964 |
| 0 2       | 54   | 0.5129  |
| 0 3       | 174  | 1.6527  |
| 1 2       | 23   | 0.2185  |
| 1 3       | 278  | 2.6406  |
| 2 0       | 15   | 0.1425  |
| 2 1       | 8    | 0.0760  |
| 2 2       | 226  | 2.1467  |
| 2 3       | 1325 | 12.5855 |
| 3 2       | 35   | 0.3324  |
| 3 3       | 458  | 4.3503  |
| 0 2 3     | 5    | 0.0475  |
| 0 3 3     | 5    | 0.0475  |
| 1 2 2     | 5    | 0.0475  |
| 1 2 3     | 90   | 0.8549  |
| 1 3 2     | 1    | 0.0095  |
| 1 3 3     | 26   | 0.2470  |
| 2 0 3     | 2    | 0.0190  |
| 2 1 3     | 6    | 0.0570  |
| 2 2 1     | 2    | 0.0190  |
| 2 2 2     | 13   | 0.1235  |
| 2 2 3     | 203  | 1.9282  |
| 2 3 2     | 3    | 0.0285  |
| 2 3 3     | 84   | 0.7979  |
| 3 1 3     | 1    | 0.0085  |
| 3 2 1     | 2    | 0.0190  |
| 3 2 2     | 1    | 0.0095  |
| 3 2 3     | 43   | 0.4084  |
| 3 3 2     | 3    | 0.0285  |
| 3 3 3     | 26   | 0.2470  |
| 0 3 2 3   | 2    | 0.0190  |
| 0 3 3 3   | 4    | 0.0380  |
| 1 2 2 3   | 9    | 0.0855  |
| 1 2 3 3   | 1    | 0.0095  |
| 1 3 2 3   | 6    | 0.0570  |
| 2 1 2 3   | 1    | 0.0095  |
| 2 2 2 3   | 1    | 0.0095  |
| 2 2 3 2   | 1    | 0.0095  |
| 2 2 3 3   | 1    | 0.0095  |
| 2 3 2 3   | 6    | 0.0570  |
| 2 3 3 3   | 4    | 0.0380  |
| 3 1 3 3   | 3    | 0.0285  |
| 3 2 2 3   | 2    | 0.0190  |
| 3 3 2 3   | 4    | 0.0380  |
| 3 3 2 2 3 | 1    | 0.0095  |

## .2 Beschreibung des Gesamtablaufs der Problemlösung

Die folgende Gesamtbeschreibung verwendet dieselbe metasprachliche Notation, die bei der Beschreibung der Operationen verwendet und am Anfang eingeführt wurde.

```
1: *MCP [MS, CODE; LK].
    LSTA [LK, LPAR; HIST, CHIQ].
    IF (*UMF [P, CHIQ, FLAG; FLAG]) GOTO 1.
    KONK [LK, SINP; KREG].
    KMPW [KREG, RINF; KWF].
    FWB [LK; WH].
    PRAE [KWF, PRFX; KWFP].
    KETT [KWFP, ZTYP; TKET].
    KMPK [TKET, KINF; KTKT].
    UEMK [TKET; UEKT].
    *SYS [UEKT, KWFP, KREG; VGRS].
2: GRUL [VGRS; RULE, PRUL].
    TRNS [KWFP; RULE, GRWF, RSTA].
    IF (*KGS [GRWF, RSTA, KWFP, VGRS; VGRS]) GOTO 2.
    *EXR [RULE, RSTA; RRUL].
    UEMG [GRWF, RRUL; UEGR, PUEG].
    KONX [UEGR, METR; KONM, PKON].
    CLSS [KONM, SCHW; NTPL, VNTP].
    IF (*KCS [NTPL, GRWF, RSTA, KWFP, KREG, VGRS; VGRS]) GOTO 2.
    WSEG [GRWF, RRUL; SEGW].
    WKST [SEGW; WKH, WKSH].
    STOP.
```

An dieser Stelle soll noch eine kurze Prosabeschreibung des gesamten Programmsystems gegeben werden. Eine graphische Beschreibung<sup>1</sup> der Operatorenfolge findet sich am Ende dieses Abschnitts.

---

<sup>1</sup>Erstellt mit dem Flußdiagrammcompiler FLOW; vgl.:  
Schönhut, J., Kurzbeschreibung des Flußdiagramm-  
Compilers FLOW, interne Programmbeschreibung,  
Erlangen 1970.

Nach Erstellung der Stichprobendaten auf Lochkarten - d.h. nach dem Ablochen des zu untersuchenden Textabschnittes - wird geprüft, ob der Stichprobenumfang ausreichend ist; ggf. wird noch mehr Text abge- locht. Ist der Stichprobenumfang ausreichend, zerlegt ein Programm den fortlaufenden Text in Wortformen, die durch Leerraum (Spatium, Blank) oder Satzzeichen begrenzt werden, ordnet das jeweilige Belegzitat und die gesamte Belegzeile zu und erstellt so die Sortierzeilen in der Form, in der sie später im Konkordanzausdruck stehen sollen, und schreibt die gesamte Sortierzeile auf ein Magnetband aus. Die auf diesem Band stehenden Daten werden anschließend durch ein Sortierprogramm in die richtige (alphabetische) Reihenfolge gebracht und sortiert auf ein anderes Magnetband ausgegeben; dabei kann die Sortierreihen- folge (alphabetische Ordnung) vorgeschrieben werden; dies ist besonders für die vorkommenden diakritischen Zeichen wichtig. Das Ergebnisband wird dann über den Schnelldrucker ausgegeben; ein weiteres Programm be- nutzt die auf dem Band stehenden Daten als Eingabe, komprimiert die mehrfach auftauchenden Wortformen auf einen Beleg; in dieser Ausgabe erscheinen in alphabetischer Reihenfolge alle Wortformen, wobei zusätzlich eine bis zu acht Zeichen lange Sigle für den Text und eine Angabe über die Vorkommenshäufig- keit in dem untersuchten Textabschnitt ausgegeben wird. Die angegebene Datei kann auch als Kartendeck gestanzt werden. Das gestanzte Kartendeck kann nach dem Beschriften mit einem Lochschriftübersetzer auch als herkömmlicher Zettelkasten verwendet werden.

Durch die eingestanzte Sigle sind Verwechslungen mit gleichartigen Kartendecks aus anderen Texten ausgeschlossen. Diese Datei dient als Eingabedatei für die folgenden Analyseprogramme.

Die nächste Programmstufe dient zum Abspalten von häufigen Praefixen; dabei wird zwischen das "Praefix" und den verbleibenden Rest der Wortform ein Trennzeichen eingefügt. Diese Maßnahme könnte auch unterbleiben, dann wären aber nach dem Segmentierprogramm umfangreiche manuelle Korrekturen in mehreren verschiedenen Dateien nötig, da die Segmentgrenze nach Praefixen in manchen Fällen von dem sonst gültigen Algorithmus zur Segmentierung abweicht. Auf die gleiche Weise werden bei zusammengesetzten Wörtern, die an den Wortbildungsgrenzen nicht den 'normalen' Bildungsgesetzen folgen, Trennzeichen an der Wortbildungsgrenze eingefügt. Dies sind reine Vorbeugungsmaßnahmen, um die Zahl der Fehlanalysen auf ein Minimum zu senken bzw. den Segmentierungs-Algorithmus nicht zu kompliziert und umfangreich (Lexikon) werden zu lassen.

Die mit diesen korrigierenden Zusätzen versehene Datei wird nun im folgenden Programm benutzt, um eine Segmentierung der Wortform in zusammenhängende Folgen von Schriftzeichen der gleichen Klasse durchzuführen. Die gefundenen Vokal- und Zeichenketten werden auf ein Band ausgegeben, sortiert und anschließend komprimiert, d.h. auf dem Ausdruck bzw. der ausgestanzten Lochkarte erscheint keine Kette mehrfach, sondern jede je einmal; sie ist mit einer Angabe über ihre Vorkommenshäufigkeit versehen. Dies allein würde an und für sich in den meisten Fällen als Ausgangsmaterial zur Erstellung eines vorläufigen Grapheminventars ausreichen; die Erfahrung hat jedoch gezeigt, daß es in manchen Fällen, besonders beim Erkennen von Allographen und Varianten, sehr hilfreich ist, wenn man gleichzeitig die Übergangshäufigkeit der Ketten und die zugehörigen Belegformen direkt, d.h. ohne zeitraubendes Suchen, zur Verfügung hat. Aus diesem Grund wurde ein Programm angeschlossen, das aus der

Wortformendatei und der Kettendatei die Übergangshäufigkeiten von einer Kette zur anderen ermittelt und zusätzlich sämtliche Belege für jeden vorkommenden Übergang zusammenstellt; dies geschieht wieder durch Sortier- und Komprimiervorgänge und über Zwischenspeicherung auf Magnetband.

An dieser Stelle setzt eine manuelle Segmentierung der gefundenen Zeichenketten unter Berücksichtigung der Umgebung ein; das Ziel dieser Arbeitsphase ist es, für die weitere Analyse ein vorläufiges Grapheminventar zu erstellen; dieses Grapheminventar legt uns in keiner Weise fest; es ist völlig flexibel und kann zu jedem späteren Zeitpunkt, wenn die weiteren Programmläufe es sinnvoll erscheinen lassen, verändert werden. Dieses Feed-back-Verfahren gestattet es, daß zur Diskussion stehende konkrete Modell iterativ der Wirklichkeit anzupassen.

Der nächste Schritt, der von dem vorigen nicht zu trennen ist, ist die Konstruktion der zu diesem Graphemsystem gehörigen Reduktionsgrammatik, d.h. wir erstellen ein Regelsystem zum Übersetzen der Originalcodierung unseres Textes (inclusive der eingefügten Trennzeichen nach Praefixen und an Wortbildungsgrenzen) in eine Codierung, in der für jedes Graphem immer nur ein Zeichen stehen darf; diese Art der Codierung verstehen wir als Graphemschreibung.

Diese Reduktionsgrammatik wird nach bestimmten Konventionen auf Lochkarten erstellt; sie dienen dann als Eingabe, für ein Programm, welches das Regelsystem in eine für die Maschine optimale Form überträgt und auf eine ständig verfügbare Datei auf Magnetplatte bringt. Das nächste Programm übersetzt nun die gesamte Wortformendatei in die Graphemschreibung (und zusätzlich noch in eine leicht lesbare Form der Graphemschreibung) und speichert diese Daten auf eine weitere Datei auf Magnetplatte. Bei systematischen Fehlern in diesen Graphemschreibungen werden

die zugehörigen Regeln im vorläufigen Graphemsystem verbessert und der Übersetzungsvorgang wiederholt. Dies geschieht iterativ so lange, bis die Fehler beseitigt sind. Auf der Basis der korrekten Graphemschreibungsdatei werden dann die Übergangshäufigkeiten von einem Graphem auf ein anderes ermittelt. Sie bilden die Basis für eine Klassenbildung auf dem System der Grapheme, die auf der Ähnlichkeit der Umgebungen von Graphemen, des Kontextes, in dem Graphem stehen, beruht. Dies geschieht über die Definition einer Metrik auf dem System der Grapheme. Damit wird es möglich, Abstände zwischen zwei Graphemen zu berechnen. Über die Einführung eines Schwellwertes gelangt man zu "ununterscheidbaren" Graphempaaren, d.h. Graphemen, deren Abstand kleiner oder gleich dem Schwellwert ist.

Die zugehörigen Klassen gewinnt man dann dadurch, daß man alle Grapheme, die voneinander ununterscheidbar sind, zu einer Klasse zusammenfaßt. Man bekommt auf diese Art eine Überdeckung des Graphemsystems. Daraus erhellt, daß es sich hier nicht um Äquivalenz-, sondern um Toleranzklassen handelt, weil das transitive Gesetz nicht notwendig gilt. Diese Toleranzklassen werden für verschiedene Schwellwerte berechnet und in Form von n-Tupeln ausgegeben, die jeweils eine Klasse enthalten.

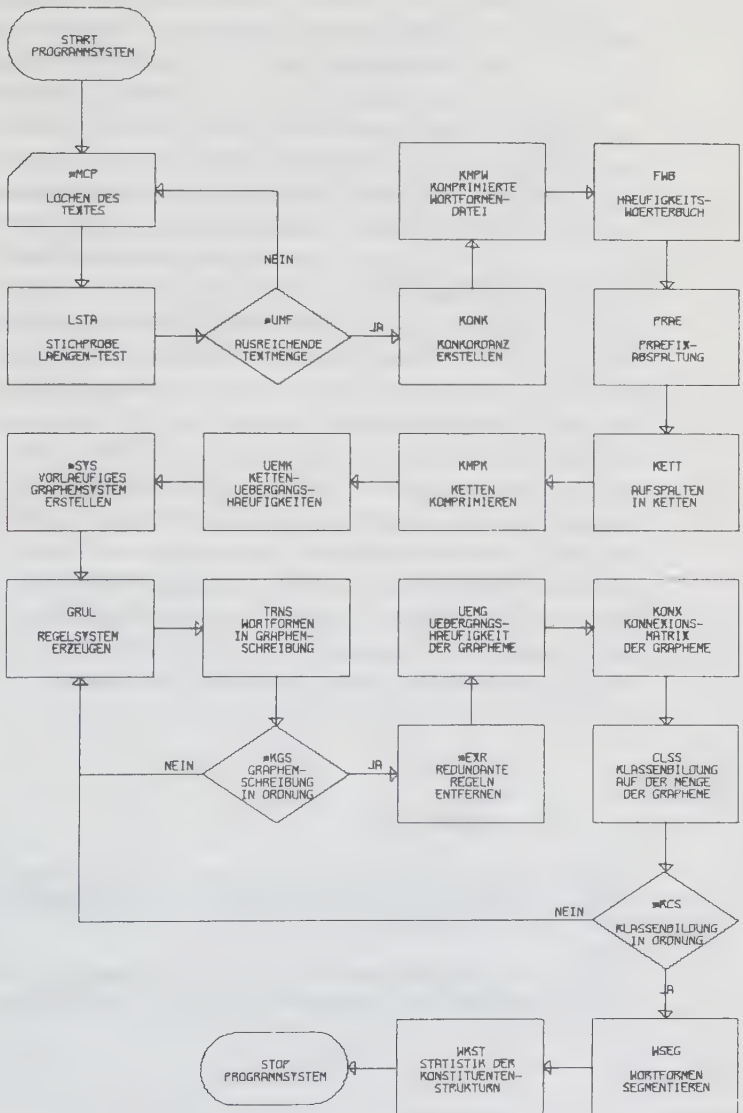
Im Anschluß an diese Auswertung besteht noch einmal die Möglichkeit, das Graphemsystem zu korrigieren; in diesem Fall wird die gesamte Programmsequenz noch einmal durchlaufen.

Der nächste Schritt geht über die Graphemebene hinaus zum Aufbau von Wortformen. Zu diesem Zweck werden die Wortformen in Segmente aufgespaltet. Ein Segment besteht aus einem Graphem der Klasse 1 (entspricht in der phonologischen Ebene einem Vokal), dem sogenannten Kern, und davor und danach einer

beliebigen Anzahl von Graphemen der Klasse 2 (phonologisch Konsonanten). Die Grapheme vor dem Kern bezeichnen wir als 2-Radikal, die Grapheme danach als 1-Radikal. Die Segmentgrenze zwischen zwei Kernen richtet sich nach der Zahl der dazwischenliegenden Grapheme. Die Segmente oder Wortkonstituenten werden je nach der Besetzung der Radikale in vier Klassen eingeteilt: bei Klasse 0 sind beide Radikal leer, bei Klasse 1 ist das 1-Radikal, bei Klasse 2 das 2-Radikal und bei Klasse 3 das 1- und das 2-Radikal besetzt. Die lineare Folge der zu einer Wortform gehörigen Segmenttypen bezeichnen wir als Wortkonstituentenstruktur. Bei der Aufspaltung der Wortformen in ihre Segmente entstehen neun Ergebnisd Dateien; eine davon wird weiter verwendet und dient zur Erstellung einer Statistik der Wortkonstituenten und Wortkonstituentenstrukturen. Diese Statistiken sind wahlweise mit Belegangabe möglich. Auf diese Art und Weise ist ein Vergleich mit den Ergebnissen anderer Texte besonders gut möglich.

Als unterstützendes Hilfsmittel wird aus dem fortlaufenden Text noch eine nach der Häufigkeit geordnete Liste aller vorkommenden Wortformen erstellt.

# GRAPHEMSYSTEM UND WORTKONSTITUENZ



### 3.3 Kurze Beschreibung der verwendeten Datenwartungs- und DatensicherungsROUTINEN

Die verwendeten Datenpflegeprogramme werden hier nur sehr knapp und pauschal beschrieben, da diese Programme stark dadurch beschränkt sind, auf welcher Anlage und unter welchem Betriebssystem sie implementiert sind. Sie hängen insbesondere stark von den Fähigkeiten des Betriebssystems zur Datei-Handhabung ab. In unserem Fall wurden alle wesentlichen Daten auf permanenten Plattendateien geführt. Diese Dateien wurden nach dem Erstellen oder nach Änderungen mit Hilfe der Standardprogramme \*FMU<sup>1</sup> bzw. SAVE einschließlich der Dateibesreibungen auf Magnetband kopiert. Auf diese Art konnte man stets die neueste Version und die Vorgängerversion einer Datei aufheben. Bei gelegentlich aufgetretenen Fehlern, die die Dateien zerstörten, konnten so mit minimalem Aufwand die Dateien wiedererstellt werden. Zur Wartung der Dateien wurden Dienstleistungsprogramme geschrieben, die Dateieinträge löschten, hinzufügten (am Ende der Datei oder in der Mitte), ersetzten oder gelöschte Dateieinträge wieder zugänglich machten. Zur Pflege der Programmbibliothek wurden die Standardprogrammsysteme GLIB und GLED<sup>2</sup> verwendet.

---

<sup>1</sup>Vgl.:

3300 3500 Computer Systems MASTER Version 3 Reference Manual, Control Data Corporation, Pub.No.60281000, 1970, Kap. 10.1.16.

<sup>2</sup>Vgl.:

3170 3300 3500 MASTER Version 3 Installation Handbook, Control Data Corporation, Pub.No.60280500G, 1971.

### 3.4 Ausführliche graphische Beschreibung der implementierten problemlösenden Programme

Im folgenden Abschnitt wird für alle unter 3.1 erläuterten Operatoren die zugehörige ausführliche Programmbeschreibung gegeben. Jede Programmbeschreibung baut sich folgendermaßen auf:

1. Liste der in FORTRAN formulierten Teilprogramme,
2. Liste der zusätzlich verwendeten Unterprogramme, die in anderen Sprachen (COMPASS, MIX-FORTRAN) geschrieben wurden, mit einer kurzen Funktionsbeschreibung,
3. Liste der Einsprungpunkte, FORTRAN-Vereinbarungen und Format-Anweisungen sowie die graphische Beschreibung für jedes einzelne in FORTRAN geschriebene Teilprogramm.

### 3.4.01 Programmbeschreibung LSTA

Das Programm LSTA besteht aus folgenden Teilprogrammen:

STATISTL (Hauptprogramm)

DUMPMTRX

RMAX

BUFFER

Zusätzlich werden folgende COMPASS-Unterprogramme verwendet:

ISCAN

Das Funktionsunterprogramm ISCAN wird aufgerufen durch

I=ISCAN(IFELD,IEQ,IDIR,IANF,IZAHL,  
SCANCHAR) oder

I=ISCAN(IFELD,IEQ,IDIR,IANF,IZAHL,  
SCANCHAR,DC).

Es dient zur Suche nach einem bestimmten Zeichen SCANCHAR in einem Feld IFELD, wobei mit der Suche in der Zeichenposition IANF bezüglich der Adresse IFELD begonnen wird und maximal IZAHL Zeichen geprüft werden. Hat IDIR den Wert 0, so wird von links nach rechts gesucht, hat es den Wert 1, so wird von rechts nach links gesucht. Wenn IEQ den Wert 0 hat, so wird auf Gleichheit mit den gesuchten Zeichen abgefragt, hat es den Wert 1, so wird nach dem ersten Zeichen ungleich den gesuchten Zeichen abgefragt. DC ist ein Zeichen, bei dessen Auftreten die Suche abgebrochen werden soll. Der Funktionswert nach dem Aufruf ist die Zahl der untersuchten Zeichen, bis eine Endbedingung erfüllt war.

MOVEC

Das Unterprogramm MOVEC (in älteren Programmen auch MOVE) wird aufgerufen durch

```
CALL MOVEC(IFELD1,IANF1,IZAHL1,IFELD2,  
           IANF2,IZAHL2).
```

Dadurch wird die Zeichenkette aus dem Feld IFELD1 beginnend mit Zeichenposition IANF1 (maximal IZAHL1 Zeichen) umgespeichert in das Feld IFELD2 beginnend in Zeichenposition IANF2, und zwar IZAHL2 Zeichen. Ist IZAHL2 größer als IZAHL1, so werden die verbleibenden Zeichen mit Leerzeichen (Blanks) gefüllt, ist IZAHL1 größer als IZAHL2, so werden nur IZAHL1 Zeichen übertragen.

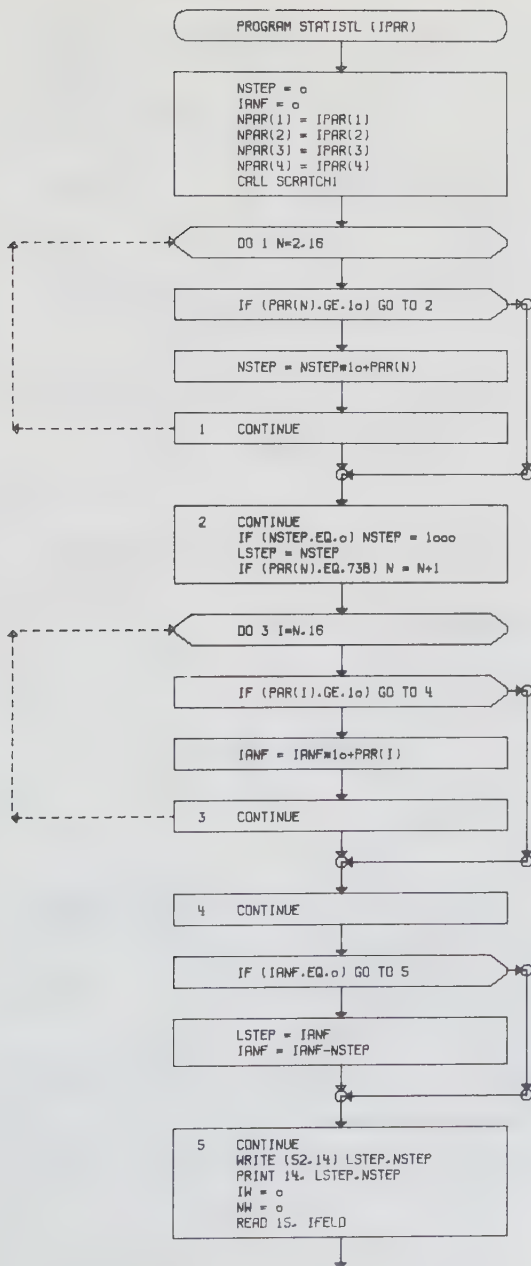
Die Unterprogramme MOVEC und ISCAN benutzen die Zeichenarithmetik (Business Data Processor, BDP) der CD 3300.

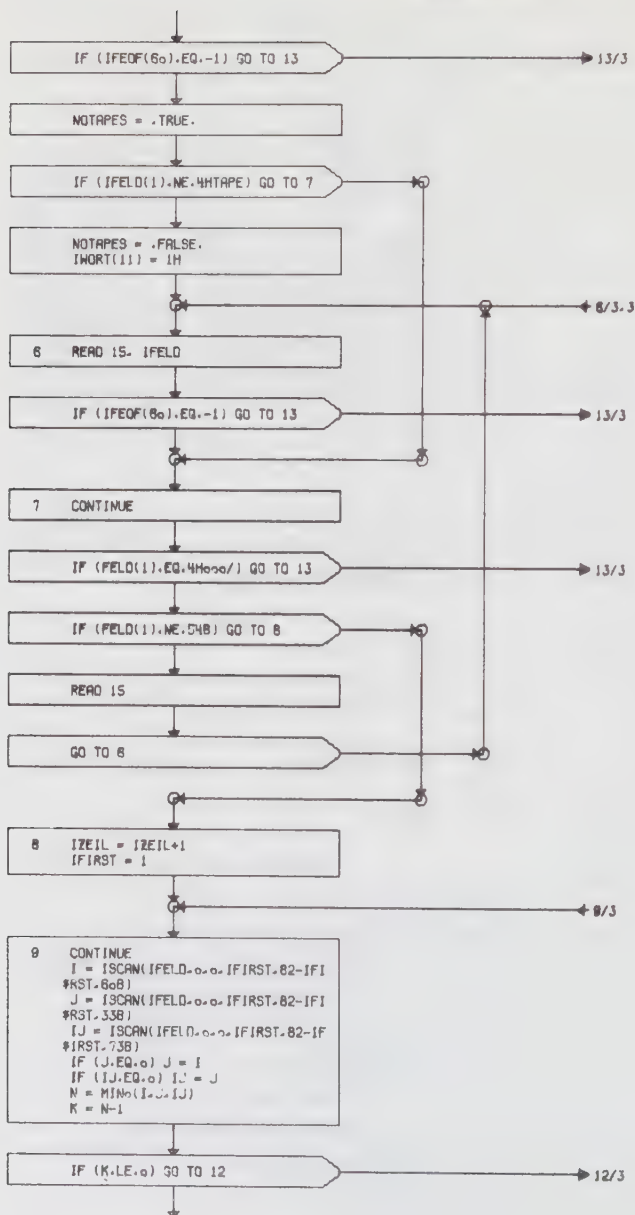
PROGRAM STATISTL (IPAR)

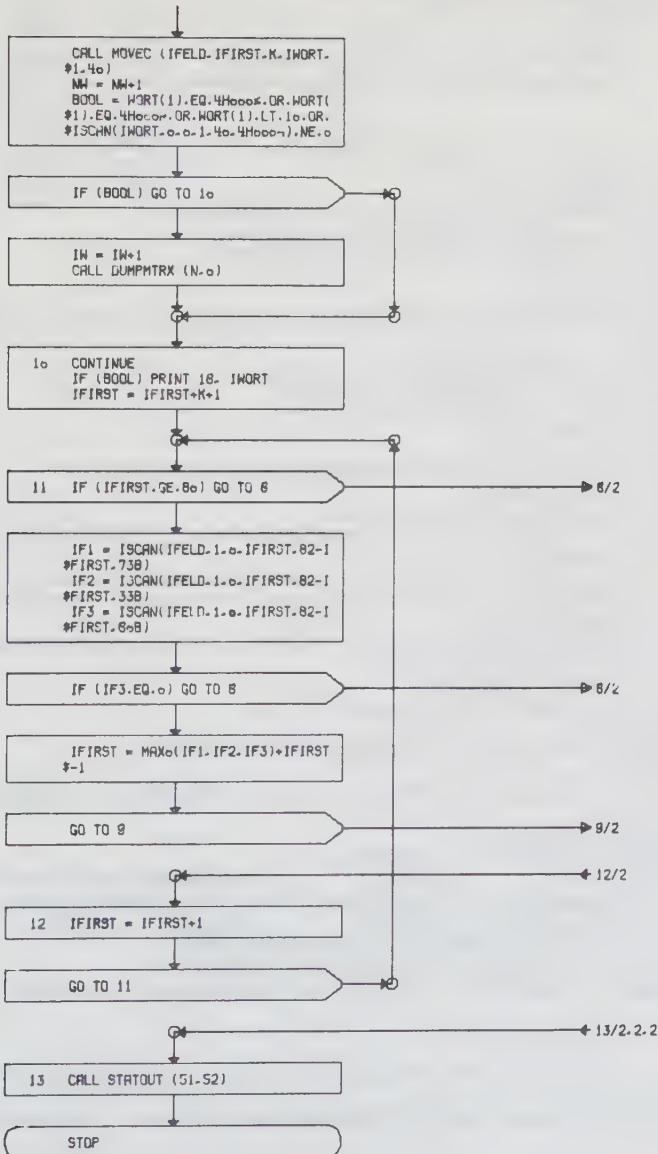
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON /TAPES/ NOTAPES
LOGICAL NOTAPES
DIMENSION IPAR(4)
CHARACTER PAR(16)
DIMENSION NPAR(2)
EQUIVALENCE (NPAR(1),PAR(1))
COMMON /STEP/ NSTEP,LSTEP,IW,NW,IANF
DIMENSION IFELD(20), IGRENZ(2)
EQUIVALENCE (IFELD(20),IGRENZ(1))
CHARACTER FELD(80),WORT(40)
COMMON /ALFA/ IWORT(10),IIIIII
EQUIVALENCE (IFELD(1),FELD(1)), (WORT(1),IWORT(1))
LOGICAL BOOL
DATA IGRENZ(2)/1H /,IZEIL/0/
```

```
14  FORMAT (20X,*ANFANGSWERT =*,I8,10X,*SCHRITTWEITE =*,I8)
15  FORMAT (20A4)
16  FORMAT (10X,10A4)
```







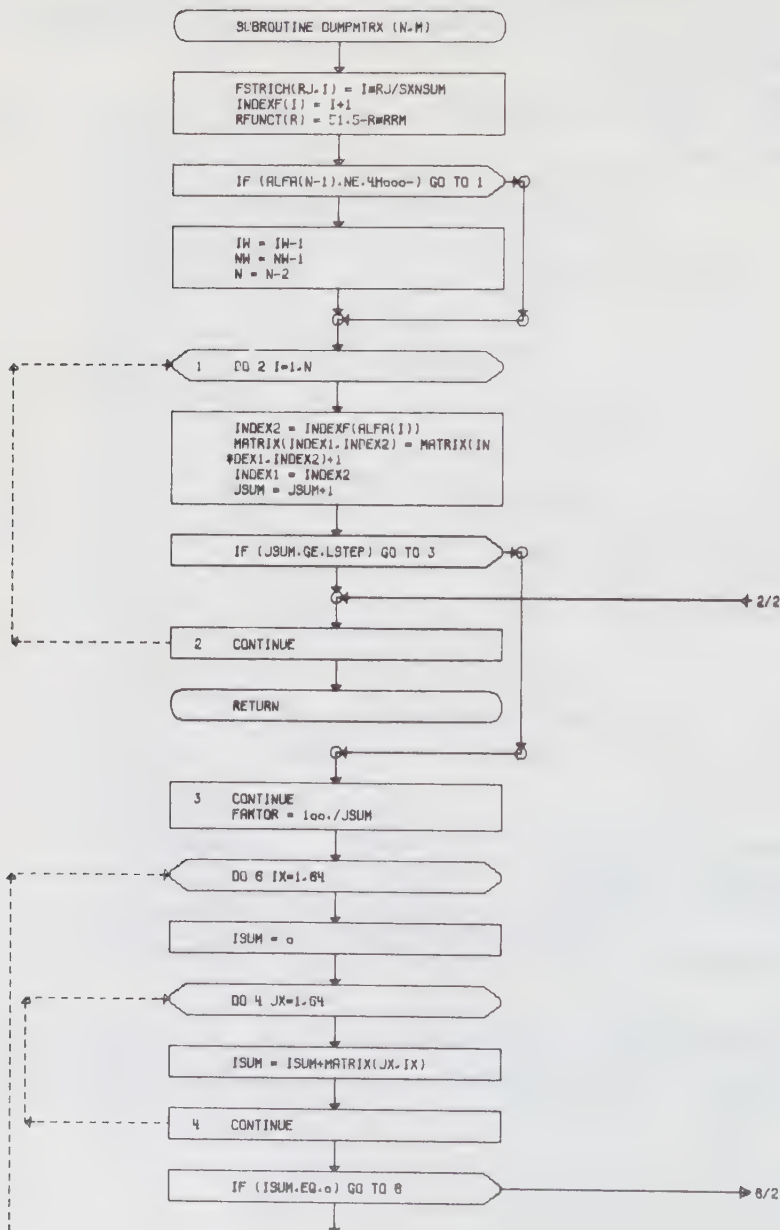
SUBROUTINE DUMPMTRX (N,M)  
ENTRY STATOUT

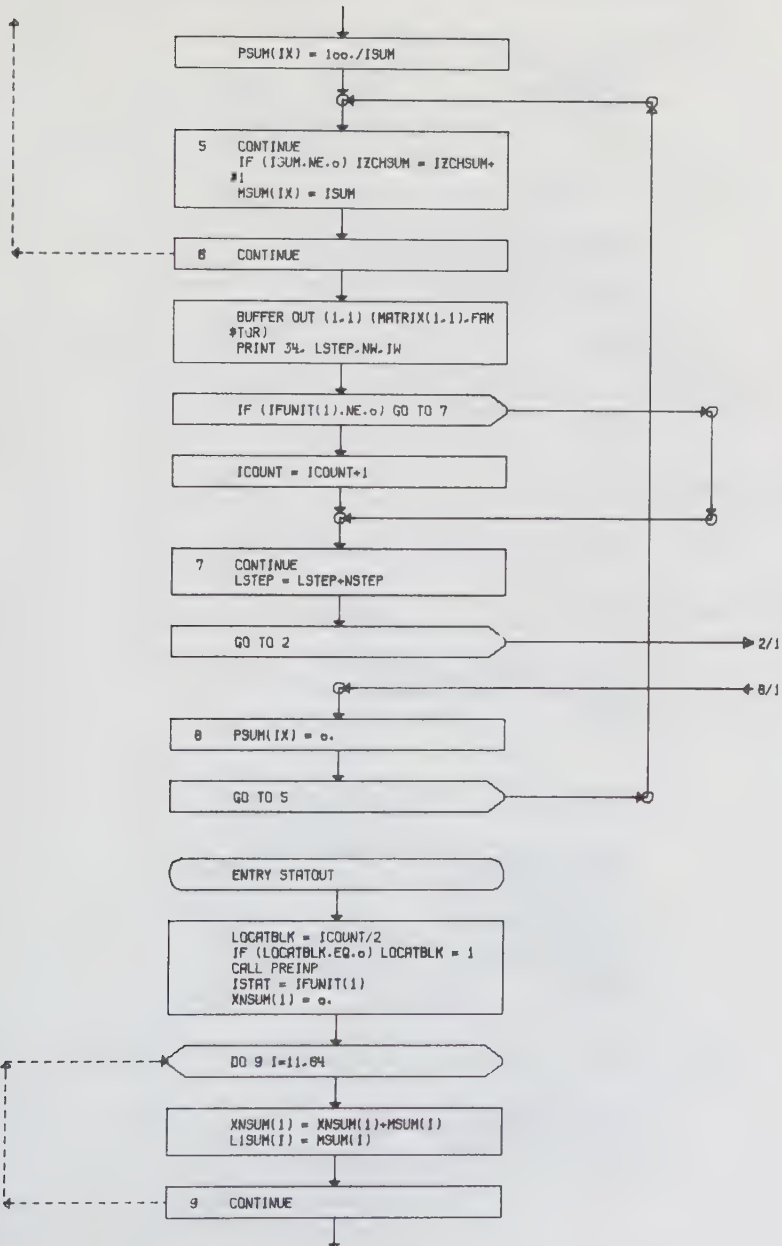
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

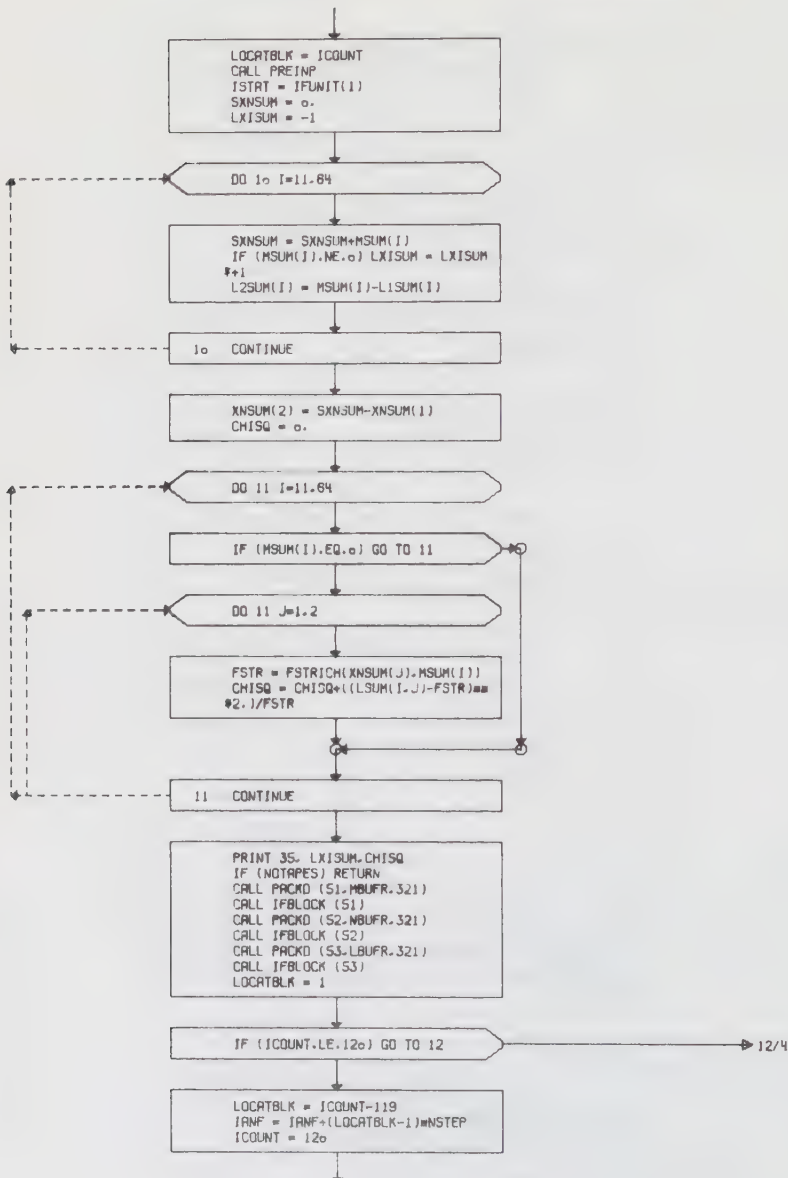
```
COMMON /TAPES/ NOTAPES
LOGICAL NOTAPES
DIMENSION NBUFR(321), MBUFR(321), LBUFR(321)
DIMENSION L1SUM(65), L2SUM(64), LSUM(64,2)
EQUIVALENCE (LSUM,L1SUM), (L1SUM(65),L2SUM)
DIMENSION XNSUM(2)
COMMON /MM/ MATRIX
DIMENSION MATRIX(64,64), MSUM(64), PSUM(64), LLL1(2), LLL2(2),
$ PLL3(2)
EQUIVALENCE (MATRIX(1,1),A)
COMMON /STEP/ NSTEP,LSTEP,IM,NW,IANF
COMMON /ALFA/ IALFA(10),IIIIII
COMMON /STATUS/ ISTAT
COMMON /PARIJ/ I,J
COMMON /PARAMET/ MATRIXIJ,MSUMI,MSUMJ,PSUMI,PSUMJ,FAKT
COMMON /LOC/ LOCATBLK
DIMENSION THAX(4), RELPROZ1(120)
EQUIVALENCE (THAX(1),PMAX), (THAX(3),PMAX1), (THAX(4),PMAX2)
CHARACTER ALFA(41)
EQUIVALENCE (IALFA(1),ALFA(1)), (MATRIX(64,64),LLL1(1)), (LLL1(2),
$MSUM(1)), (MSUM(64),LLL2(1)), (LLL2(2),PSUM(1)), (PSUM(64),PLL3(1)
$), (PLL3(2),FAKTOR)
DIMENSION ID(2), MELEMENT(120), MELSUM(120,2), PELPROZ(120,2),
$ PYPROZ(120), IDISPLAY(34,51)
CHARACTER DISPLAY(136,51)
EQUIVALENCE (IDISPLAY(1,1),DISPLAY(1,1))
DIMENSION R(10)
DIMENSION BLOCK(17,56), TEXTL(2), UBLOCK(17,3)
CHARACTER CBLOCK(136,56)
EQUIVALENCE (BLOCK(1,4),IDISPLAY(1,1)), (BLOCK(1,1),CBLOCK(1,1),
$UBLOCK(1,1))
DIMENSION IBLOCK(34,56)
EQUIVALENCE (BLOCK(1,1),IBLOCK(1,1))
DATA UBLOCK/8H1,2*8H,80HRELATIVE HAEUFIGKEIT VON
$VOR ALS FUNKTION DER TEXTLAENGE (ZAHL DER ZEICHEN),7*8H
$ ,3*3H***** ,3H***** ,9*8H ,8H100 % = ,11*8H /
DATA ZAHLDER/8HZAHL DER/,TEXTL/16HTEXTLAENGE /
DATA R/.1,.2,.5,1.,2.,5.,10.,20.,50.,100./
DATA JSUM/0/,INDEX1/49/,MSUM/64*0/,PSUM/64*0./,ICOUNT/0/
```

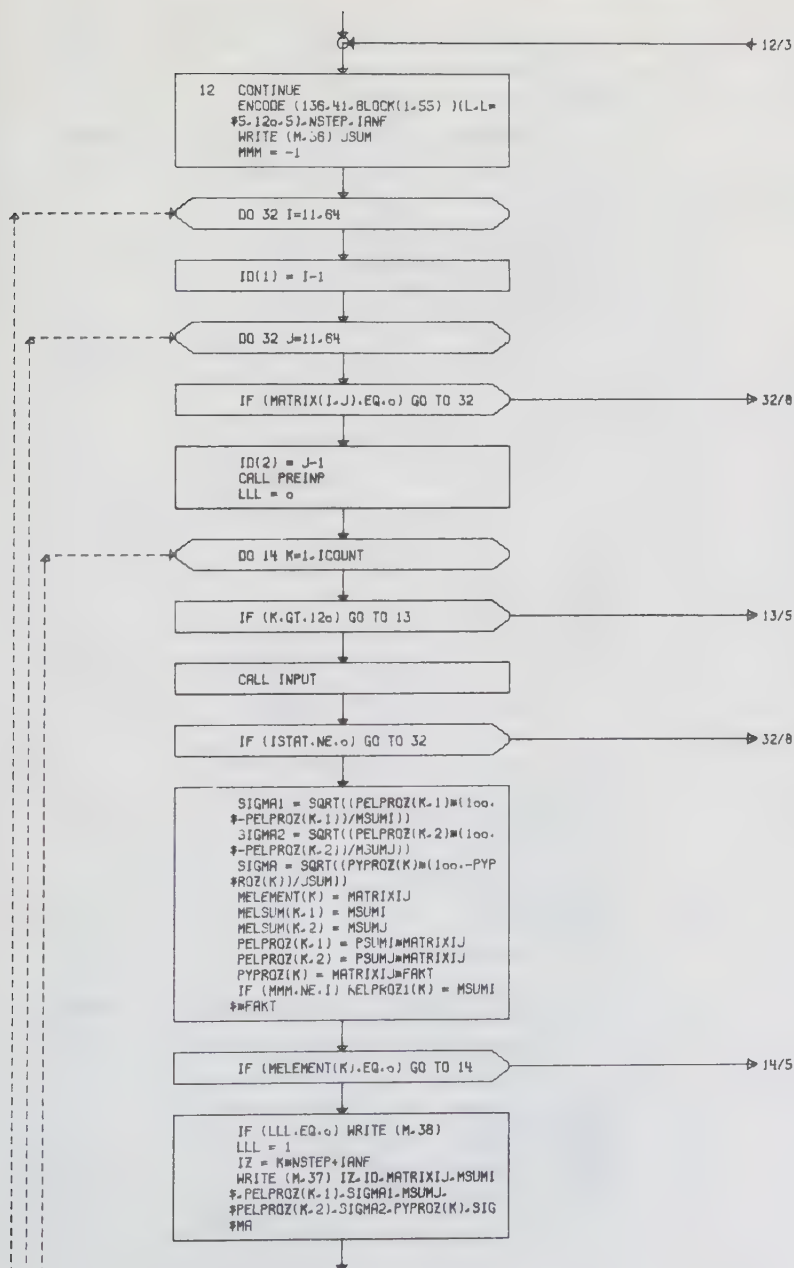
- 34 FORMAT (10X,I10,1X,7HZEICHEN,3X,I10,1X,33HWORTFORMEN, DAVON BERUEC  
\$KSICHTIGT,I10)
- 35 FORMAT(\*1\*,5X,\*ZAHL DER FREIHEITSGRADE = \*.I3\* CHI-QUADRAT =\*,  
\$F8.4)

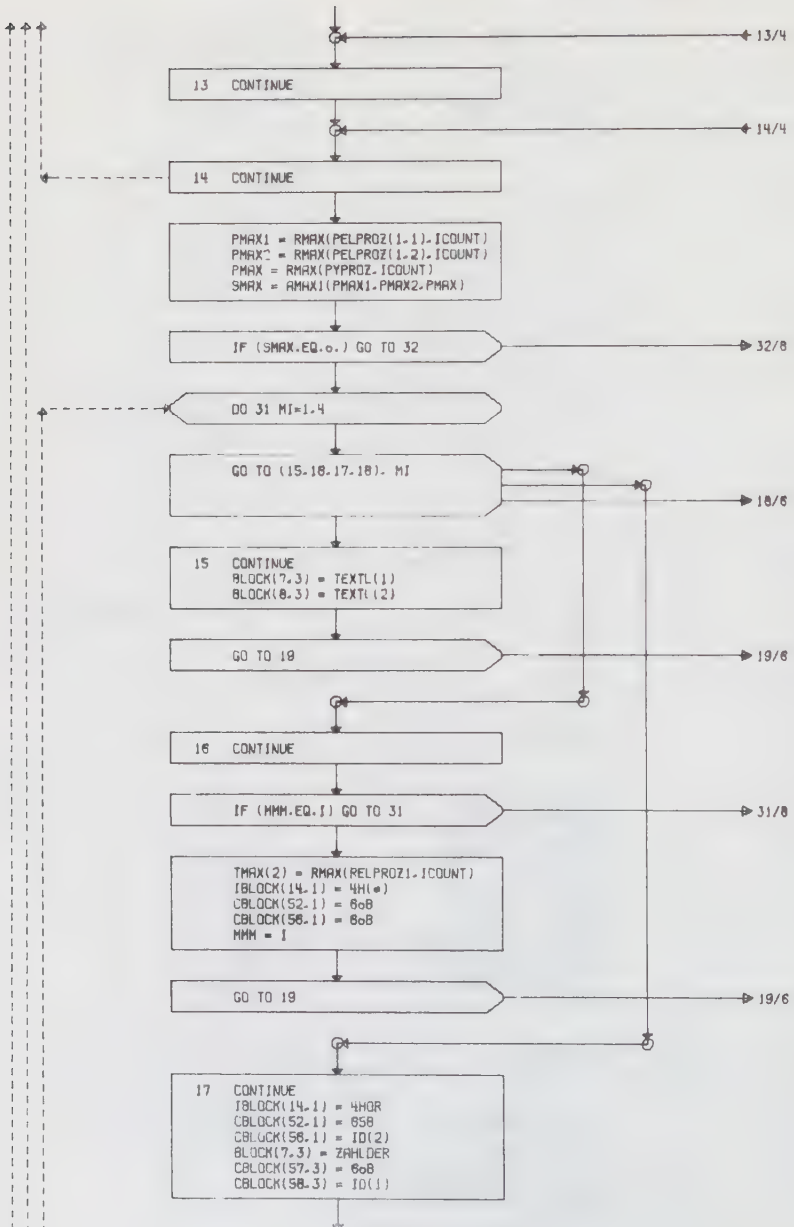
```
36  FORMAT (20X,30HZ AHL DER VERARBEITETEN ZEICHEN,1X,I10)
37  FORMAT (15X,I8,5X,R1,1X,R1,2X,I8,2(2X,I8,2(2X,F8.4)),2(2X,F8.4))
38  FORMAT (1H1,16X,6H LAENGE,3X,7H ZEICHEN,1X,7H ELEMENT,3X,7H MEL SUM1,2X
$,8H REL PROZ1,4X,6H SIGMA1,3X,7H MEL SUM2,2X,8H REL PROZ2,4X,6H SIGMA2,3X,
$8H REL (GES),5X,5H SIGMA)
39  FORMAT (F8.2,4H % -)
40  FORMAT (34A4)
41  FORMAT (16X,24I5/30X,8HX-WERT *,I8,2H +,I8,33H = ZAHL DER VERARBEI
$TETEN ZEICHEN)
42  FORMAT (34A4)
```

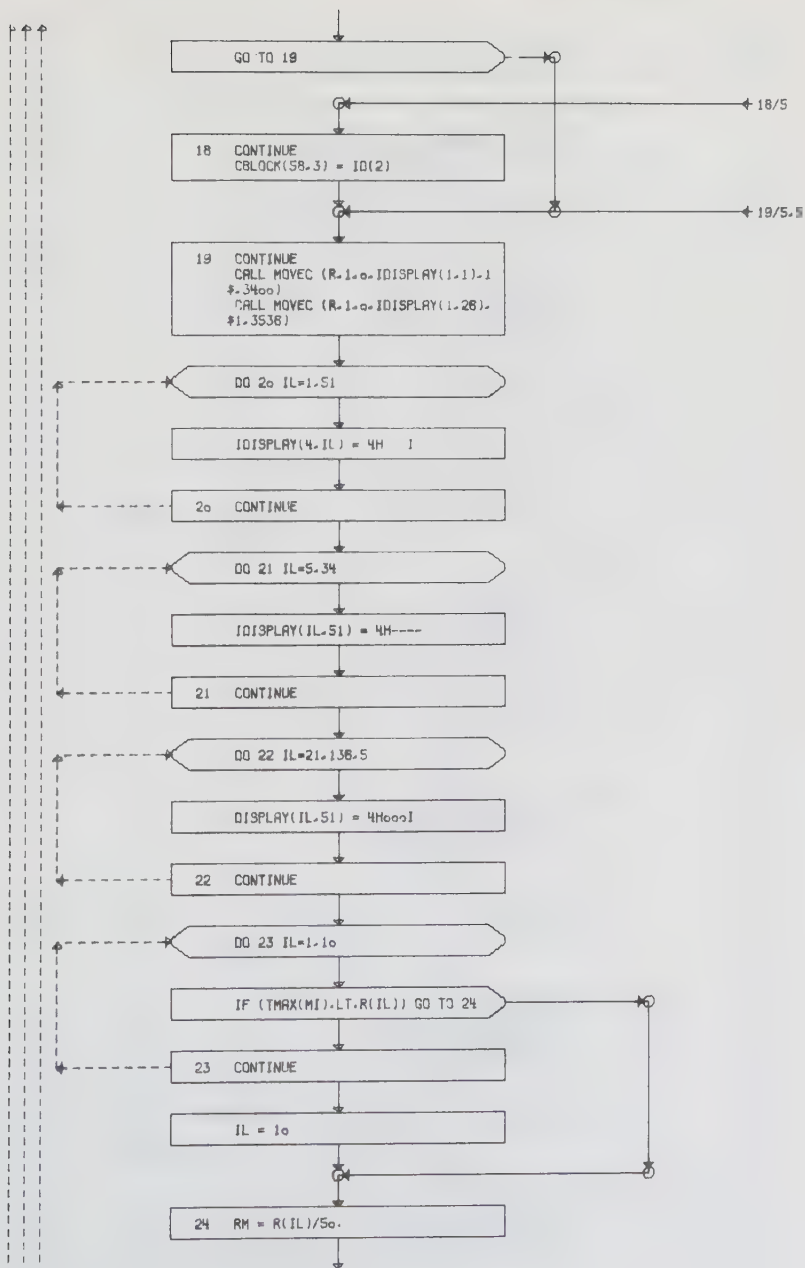


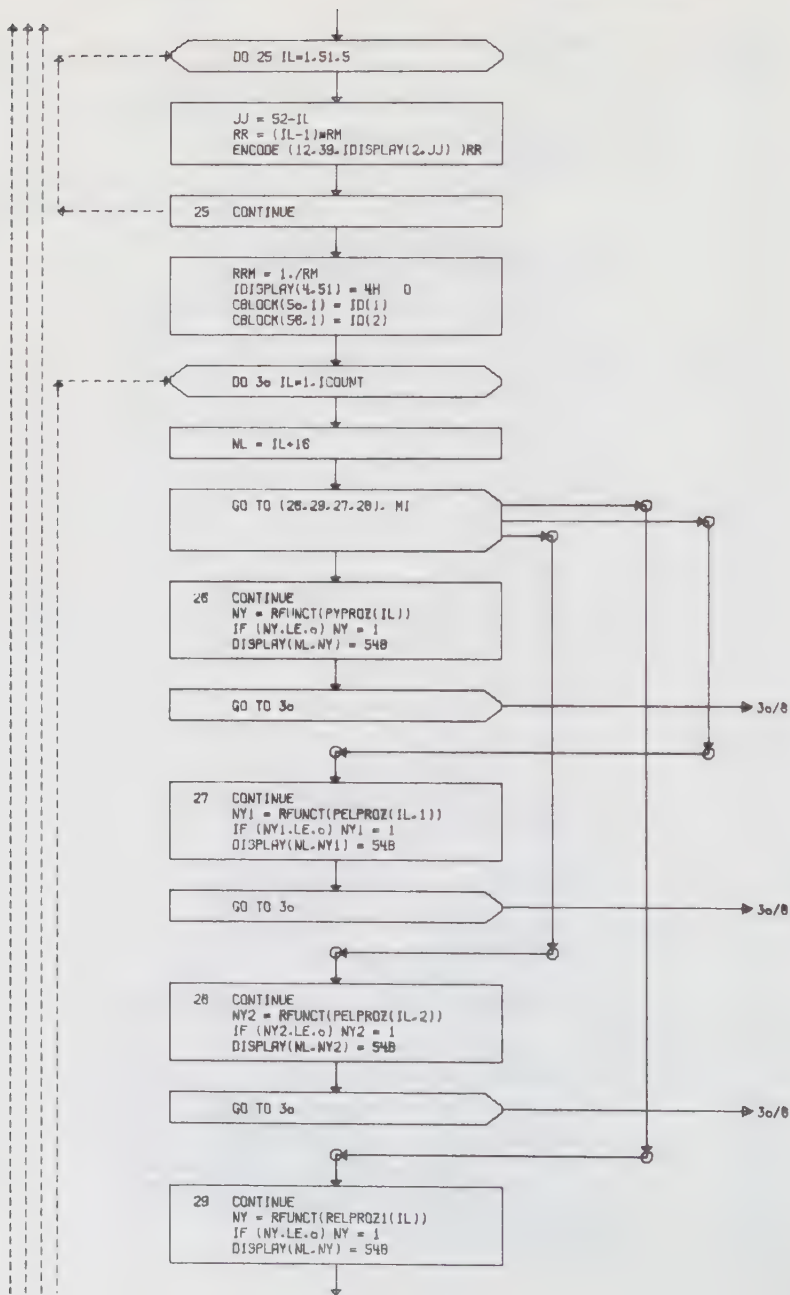


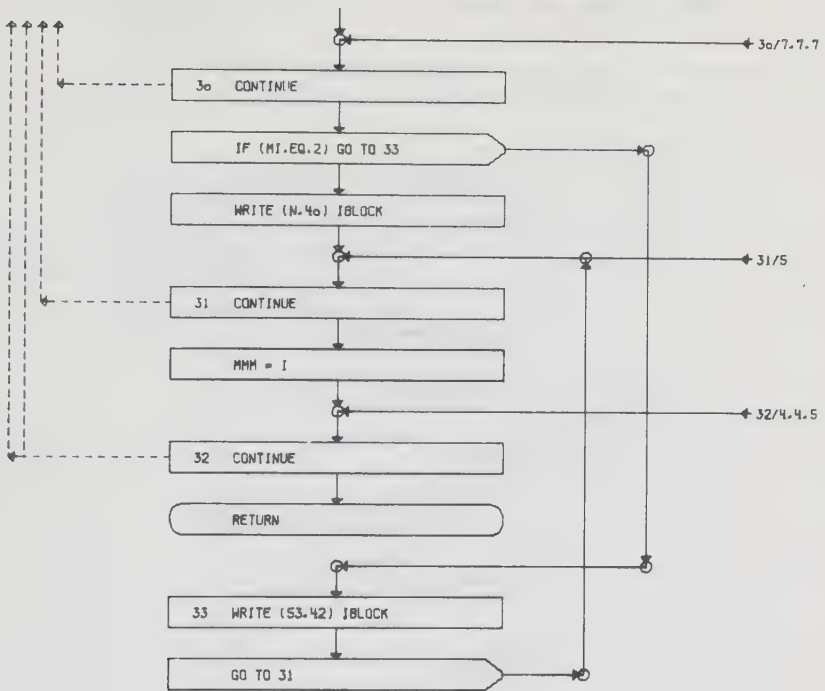










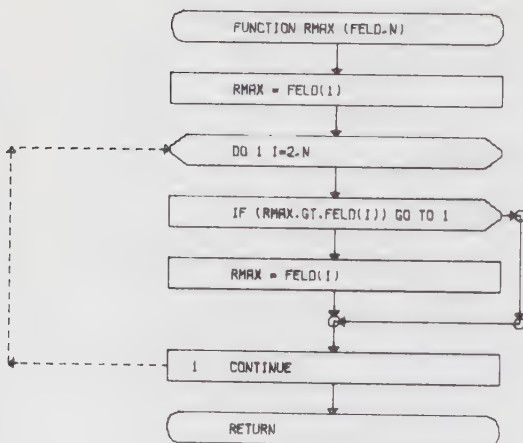


FUNCTION RMAX (FELD,N)

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

DIMENSION FELD(1)

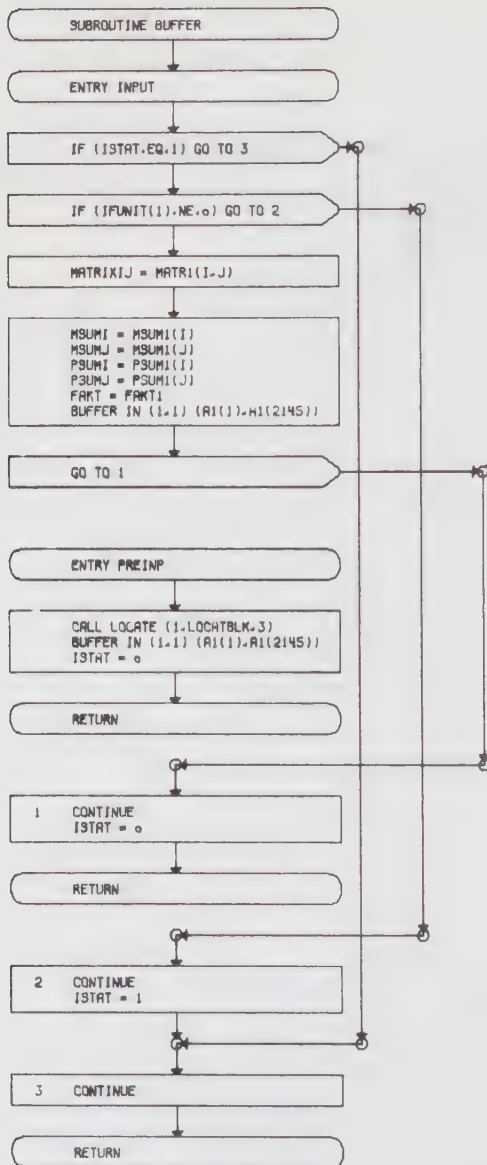
FUNCTION RMAX



```
SUBROUTINE BUFFER  
ENTRY INPUT  
ENTRY PREINP
```

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON /PARAMET/ MATRIXIJ,MSUMI,MSUMJ,PSUMI,PSUMJ,FAKT  
COMMON /PARIJ/ I,J  
COMMON /MM/ MATR1(64,64),MSUM1(64),PSUM1(64),FAKT1  
COMMON /STATUS/ ISTAT  
COMMON /LOC/ LOCATBLK  
DIMENSION A1(2145)  
EQUIVALENCE (A1,MATR1)
```



### 3.4.02 Programmbeschreibung KONK

Das Programm KONK besteht aus folgenden Teilprogrammen:

DTAUFB (Hauptprogramm)

AUFBER

Zum Sortieren der Ausgabedatei wird das unter dem Betriebssystem MASTER verfügbare Standardsortierprogramm SORT<sup>1</sup> verwendet. Nach dem Sortieren wird die Datei mit Hilfe der Standardkopierroutine des Betriebssystems XFER auf dem Schnelldrucker ausgegeben.

---

<sup>1</sup>Vgl. dazu:

3300 3500 Computer Systems Mass Storage SORT/  
MASTER; Control Data Corporation, Pub.No.60206600,  
1967.

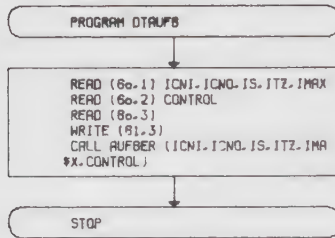
PROGRAM DTAUF8

LOGICAL CONTROL

```
1  FORMAT (I2)
2  FORMAT (1L1)
3  FORMAT (1H )
```

PROGRAM DTAUF8

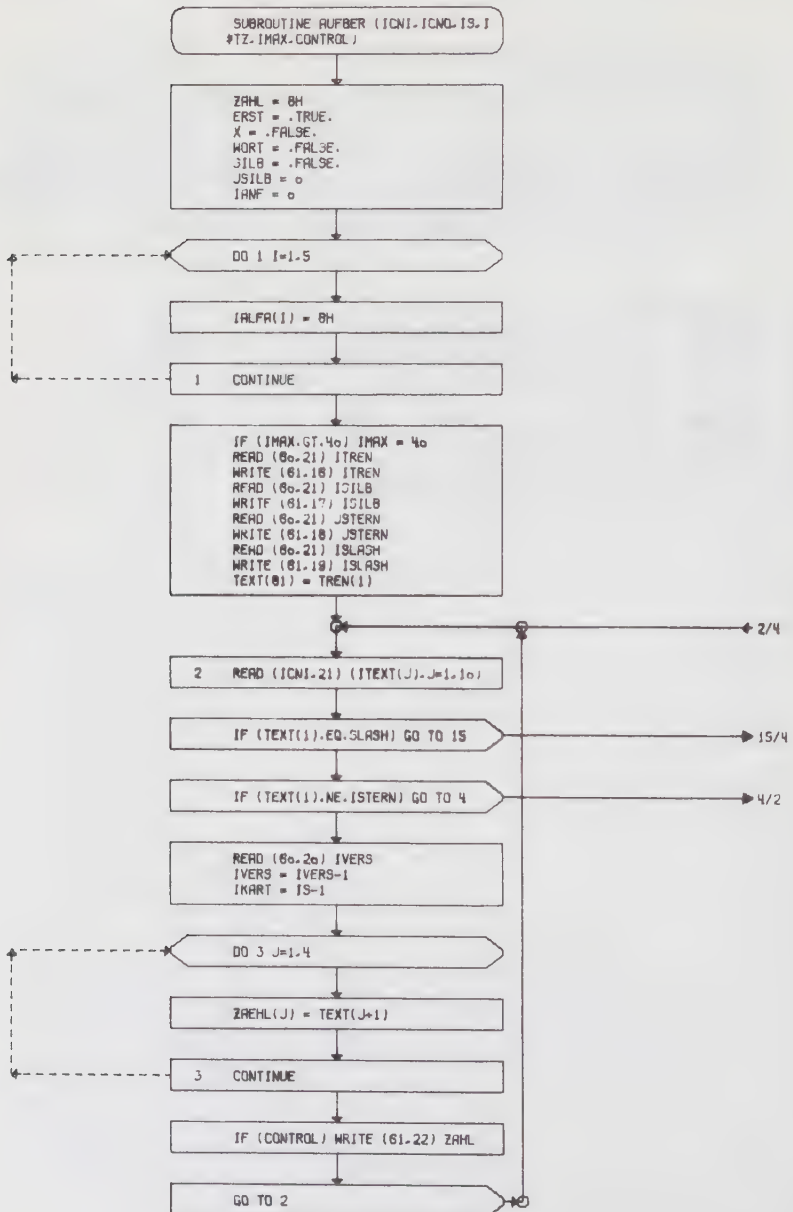
1

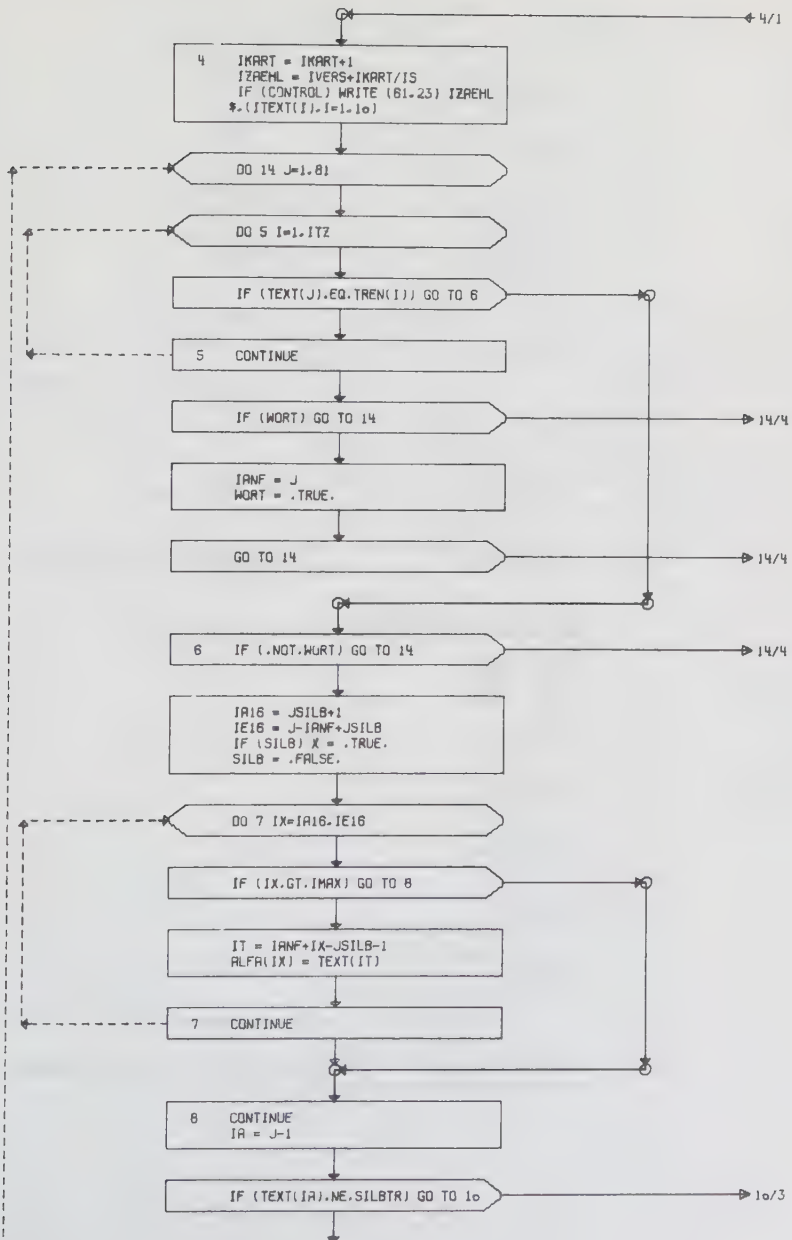


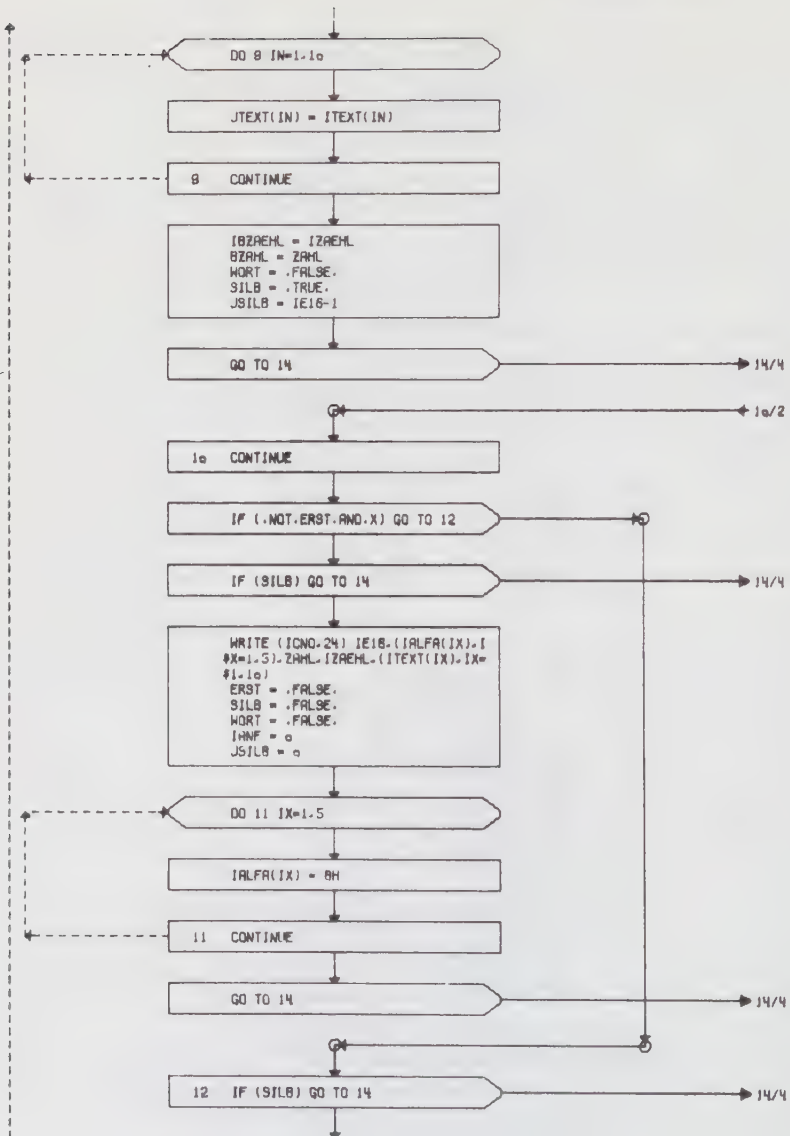
SUBROUTINE AUFBER (ICNI,ICNO,IS,ITZ,IMAX,CONTROL)

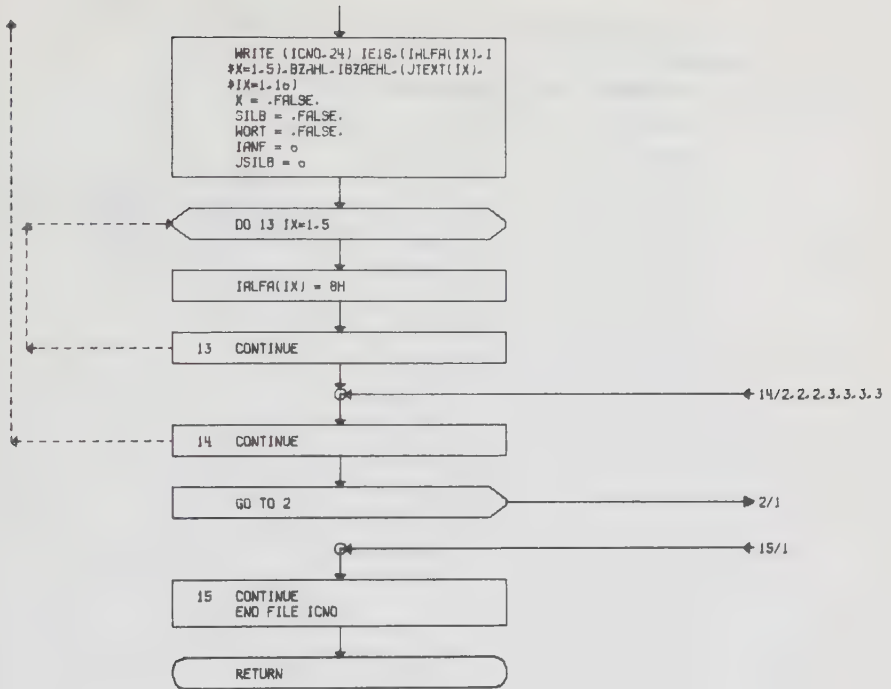
LOGICAL CONTROL,WORT,SILB,ERST,X  
 INTEGER ZAHL,BZAHL  
 CHARACTER TEXT,BZAEHL,ZAEHL,TREN,ALFA,SLASH,ISTERN,SILBTR  
 DIMENSION BZAEHL(8), ZAEHL(8), TEXT(81), TREN(8), ALFA(40), ITEXT  
 \$(11)  
 DIMENSION IALFA(5), JTEXT(10)  
 EQUIVALENCE (BZAEHL(1),BZAHL), (ZAEHL(1),ZAHL), (ITEXT(1),TEXT(1))  
 EQUIVALENCE (ITREN,TREN(1)), (ISTERN,JSTERN), (ISLASH,SLASH)  
 EQUIVALENCE (ALFA(1),IALFA(1)), (ISILB,SILBTR)

16 FORMAT (1H ,8HTRENNZEI,8HCHE ,8H ,A8)  
 17 FORMAT (1H ,8HSILBENTR,8HENNER ,8H ,A8)  
 18 FORMAT (1H ,8HSEITENST,8HEUERKART,3HE ,A8)  
 19 FORMAT (1H ,8HTEXTENDE,8HMARKE ,8H ,A8)  
 20 FORMAT (I7)  
 21 FORMAT ((10A8))  
 22 FORMAT (1H1,20X,1A4)  
 23 FORMAT (1H ,I5,5X,10A8)  
 24 FORMAT (I2,1X,5A8,1X,1A4,I5,2X,(10A8))









#### 3.4.03 Programmbeschreibung KMPW

Das Programm KMPW besteht aus folgenden Teilprogrammen:

DATAKMPR (Hauptprogramm)

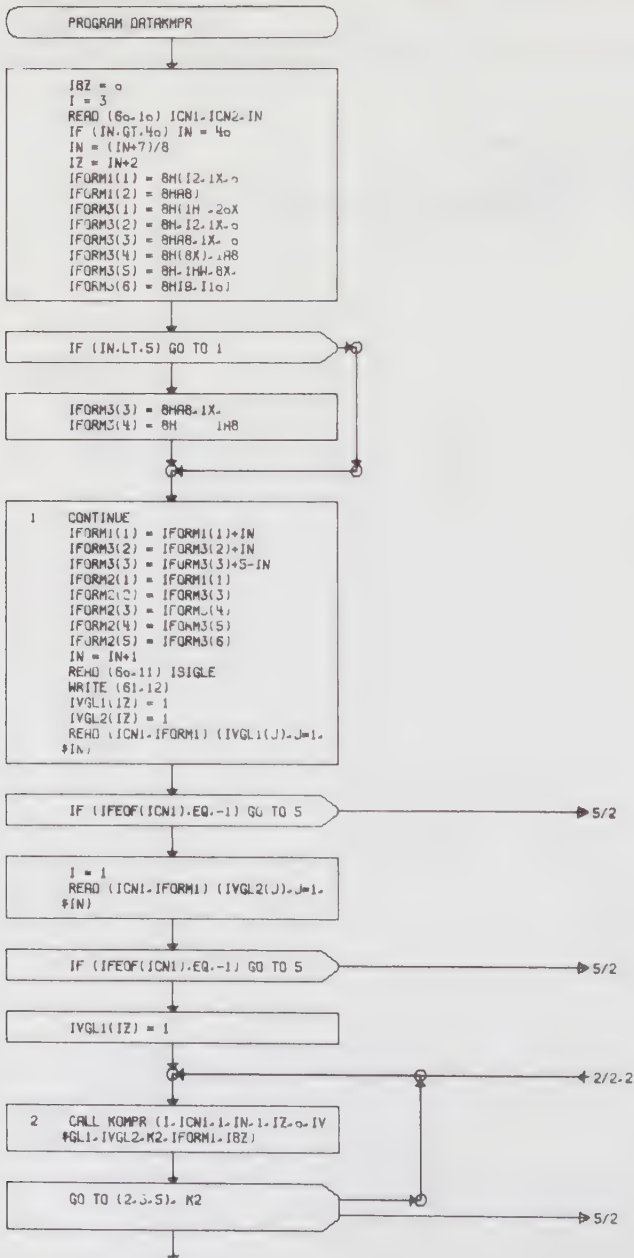
VGLSTR

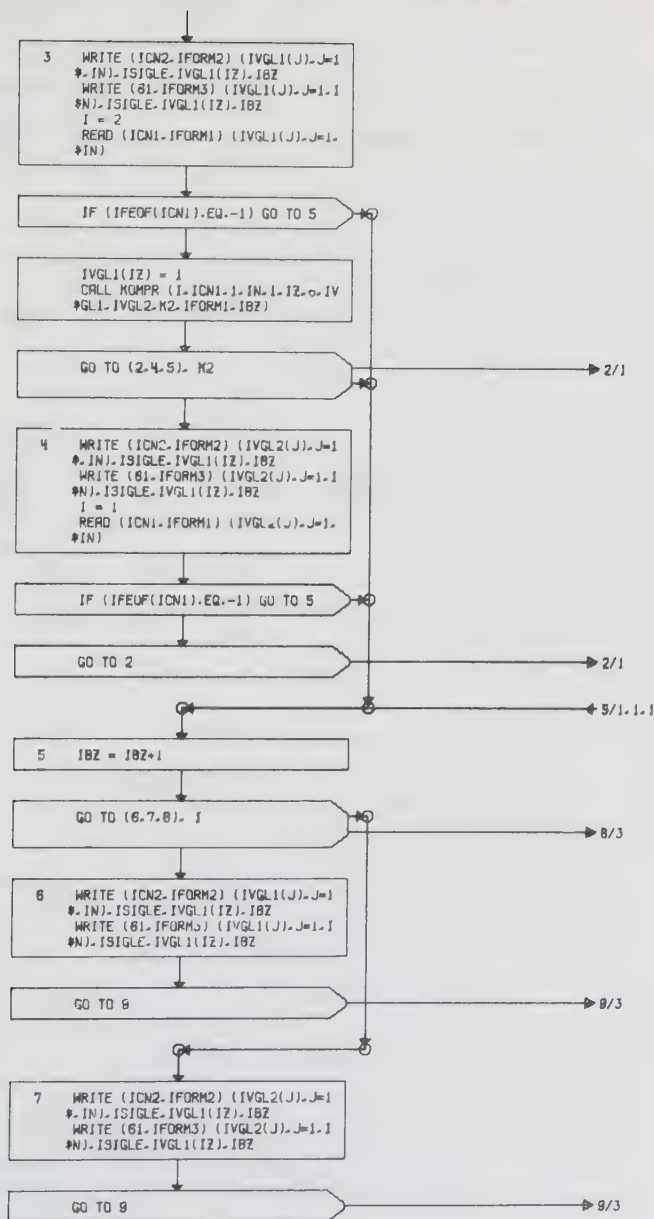
KOMPR.

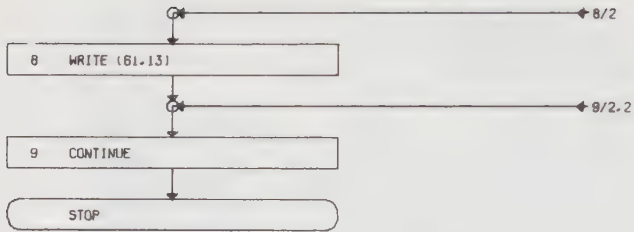
PROGRAM DATAKMPR

DIMENSION IVGL1(7), IVGL2(7), IFORM1(2), IFORM2(5), IFORM3(6)

```
10  FORMAT (3(I2,1X))
11  FORMAT (1A8)
12  FORMAT (1H1,/1HR)
13  FORMAT (20X,18HKEINE EINGABEDATEN)
```





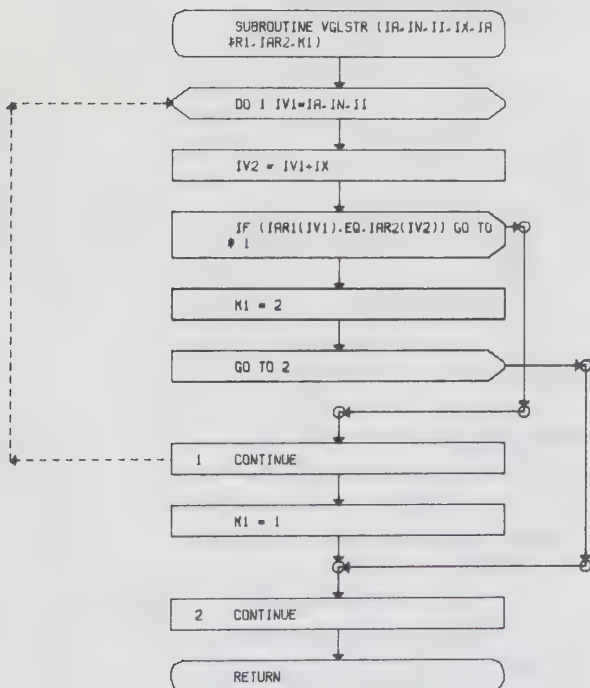


SUBROUTINE VGLSTR (IA,IN,II,IX,IAR1,IAR2,K1)

DIMENSION IAR1(1), IAR2(1)

SUBROUTINE VGLSTR

1

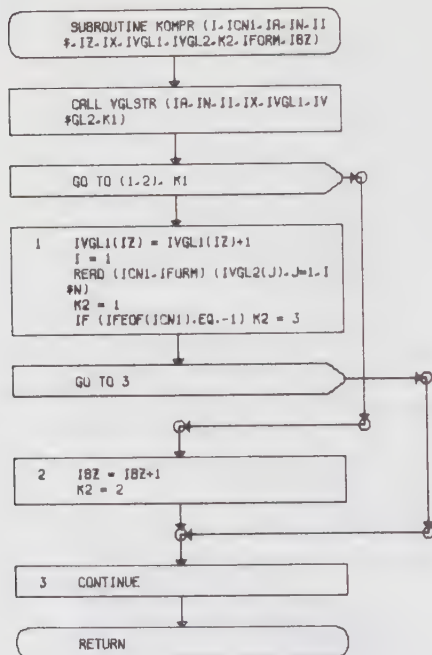


SUBROUTINE KOMPR (I,ICN1,IA,IN,II,IZ,IX,IVGL1,IVGL2,K2,IFORM,IBZ)

DIMENSION IVGL1(1), IVGL2(1), IFORM(1)

SUBROUTINE KOMPR

1



### 3.4.04 Programmbeschreibung WH

Das Programm WH enthält ein FORTRAN-Programm:

HAEUF<sup>1</sup> (Hauptprogramm).

Zusätzlich werden folgende COMPASS-Unterprogramme verwendet:

MOVEC (siehe 3.4.01)

ICMPR

Das Funktionsunterprogramm ICMPR wird aufgerufen durch

I=ICMPR(J,IFELD1,IANF1,IZAHL1,IFELD2,  
IANF2,IZAHL2).

Damit werden die Zeichenketten in IFELD1 und IFELD2, beginnend mit Zeichenposition IANF1 bzw. IANF2 von der Länge IZAH1 bzw. IZAH2 auf Gleichheit geprüft. J enthält die Zahl der verglichenen Zeichen nach der Ausführung. Ist der Wert von IZAH1 nicht gleich dem von IZAH2 und ist bis zum kleineren der beiden Werte Gleichheit der Zeichenkette festgestellt, so wird geprüft, ob die längere Zeichenkette hinten mit Leerzeichen gefüllt ist. Ist dies nicht der Fall, so werden die Ketten nicht als gleich erkannt. Bei Gleichheit ist der Funktionswert 0, sonst ist er 1 (IFELD1 größer IFELD2) oder 2 (IFELD1 kleiner IFELD2). Dieses Programm benutzt die Zeichenarithmetik (BDP).

SORT

Das Unterprogramm SORT wird aufgerufen durch

CALL SORT(IFELD,IZAHL,IRECZAHL,IZWI).

Dabei ist IFELD das zu sortierende Datenfeld, IZAHL die Länge der Sortiereinheiten in Zeichen, IRECZAHL die Zahl der zu sortierenden Einheiten und IZWI ein Feld, das ebenso

lang sein muß, wie eine zu sortierende Einheit, und zum zyklischen Vertauschen beim Sortieren dient. Die Sortierreihenfolge ist die Reihenfolge des Alphabets.

---

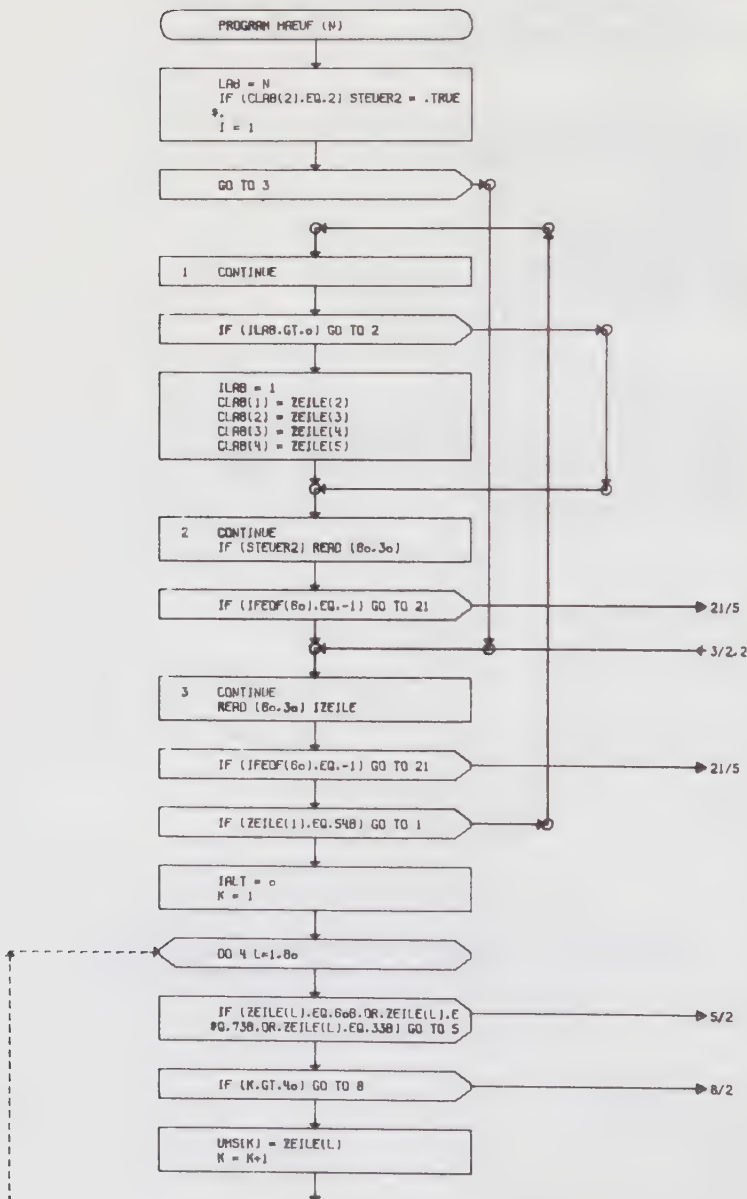
<sup>1</sup>Dieses Programm ist die umgeschriebene Version eines Programmes von G. Koller.

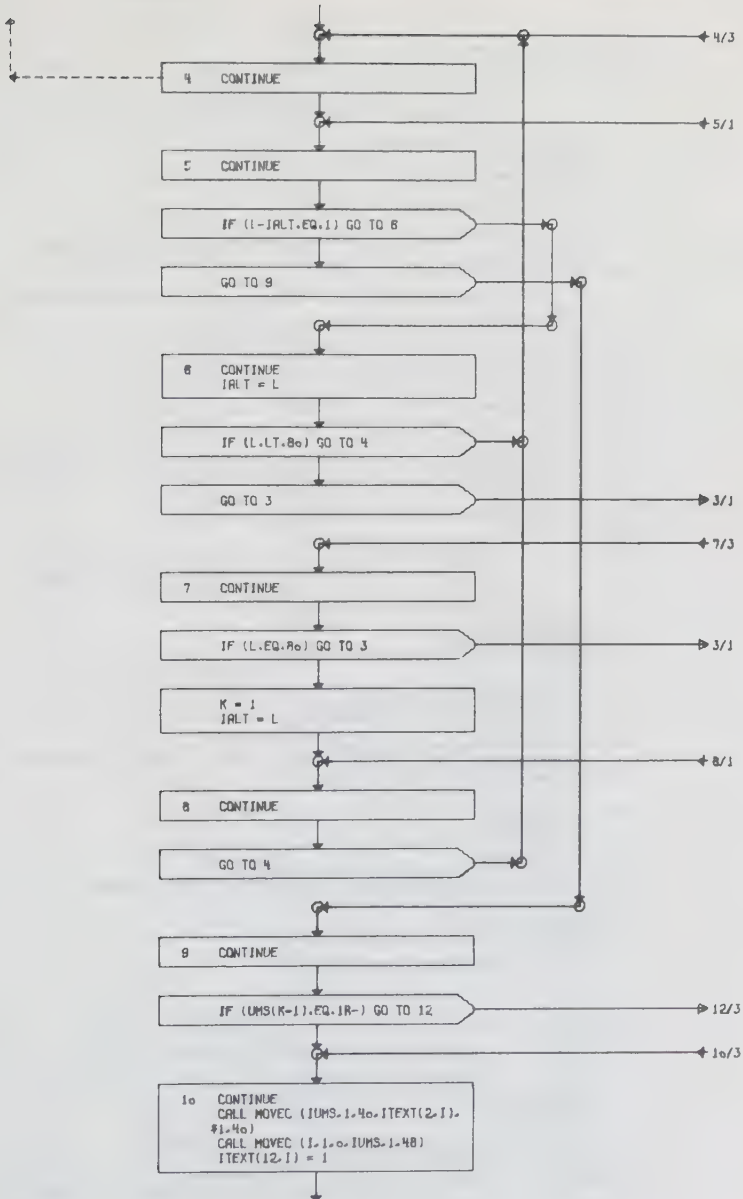
PROGRAM HAEUF (N)

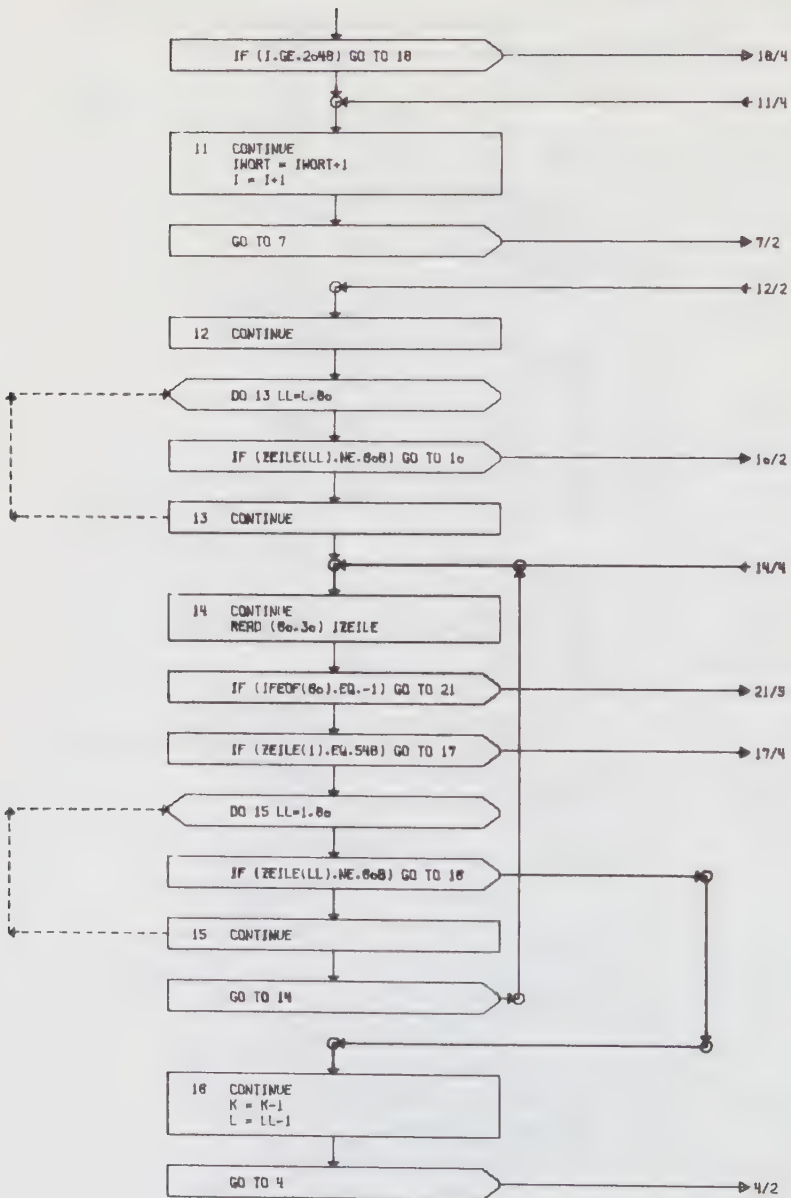
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

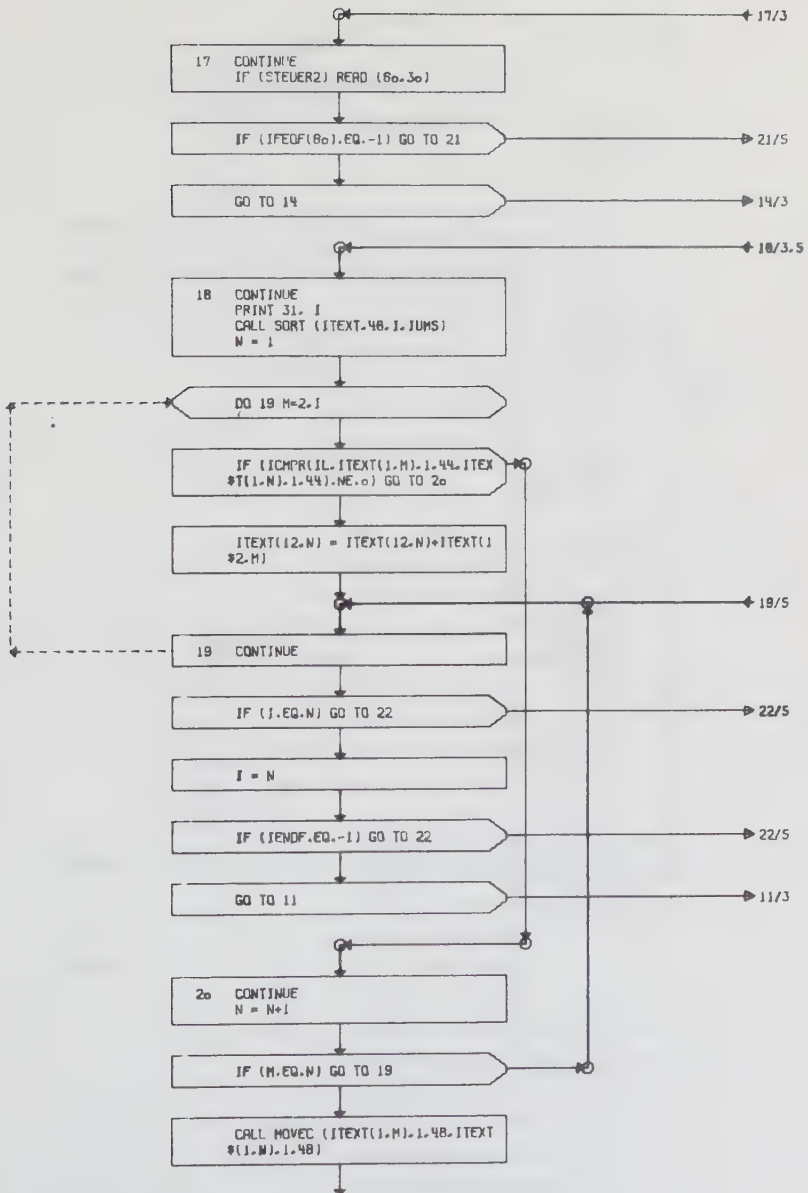
```
DIMENSION ITEXT(12,2048), IZEILE(20), IUMS(12)
CHARACTER ZEILE(80),UMS(48)
EQUIVALENCE (ZEILE,IZEILE), (IUMS,UMS)
COMMON ITEXT
EQUIVALENCE (LAB,CLAB)
CHARACTER CLAB(4)
LOGICAL STEUER2
DATA STEUER2/.FALSE./
DATA IUMS/12*1H /
DATA MP/50/
```

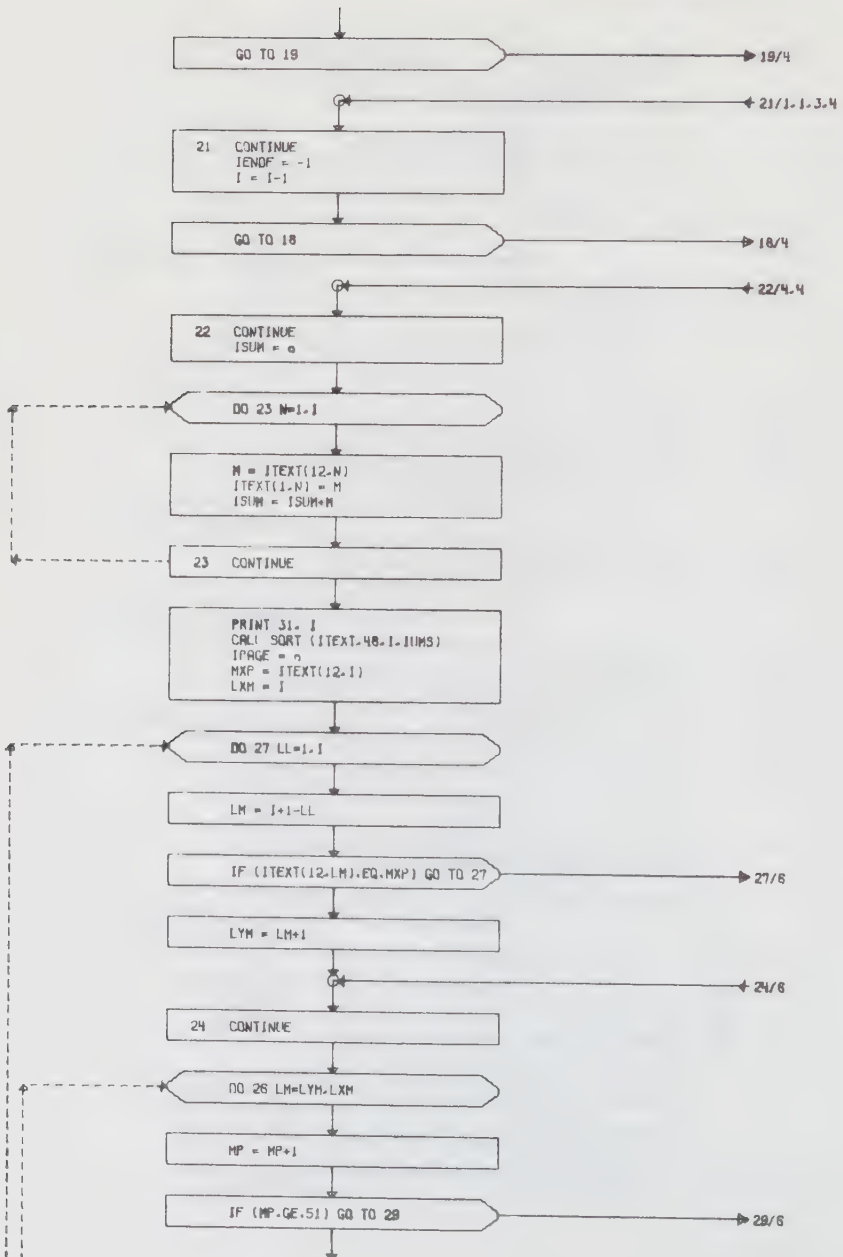
```
30  FORMAT (20A4)
31  FORMAT (5X,10(1H*),*SORT-AUFRUF,I=*,I5)
32  FORMAT (5X,I5,*,*,1X,10A4,I8,F8.3)
33  FORMAT(*1*,5X,*BEI *,I5,* LAUFENDEN WORTFORMEN KAMEN *,I5,* VERSCH
$IEDENE WORTFORMEN VCR.*)
34  FORMAT (*1*/*-*/*-*)
```

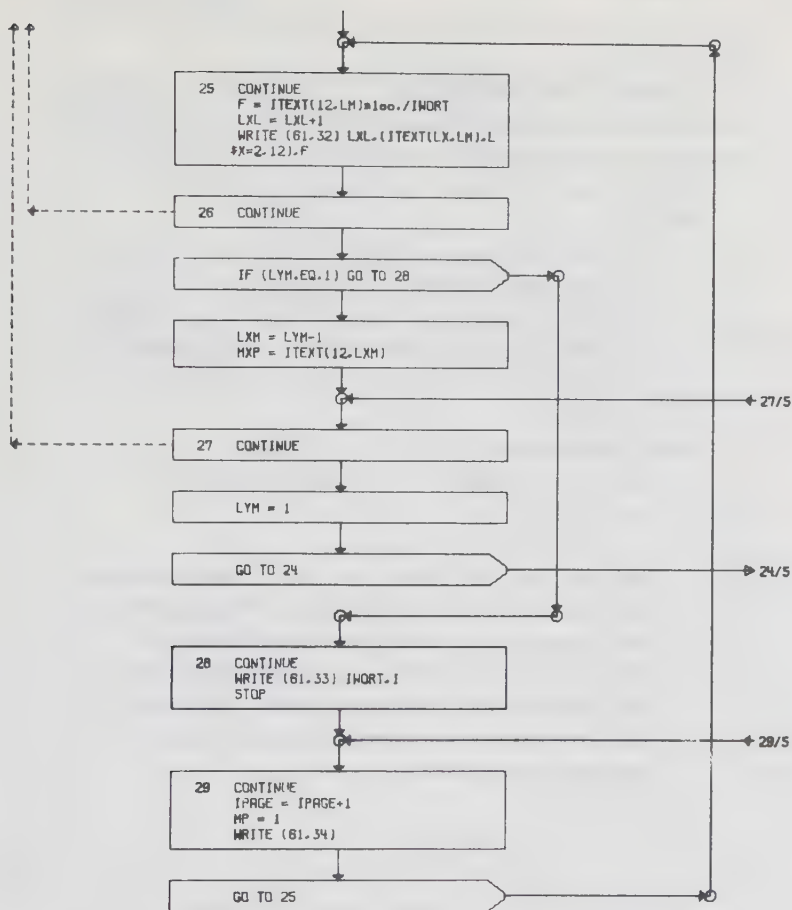












### 3.4.05 Programmbeschreibung PRAE

Das Programm PRAE besteht aus folgenden Teilprogrammen:

PRAEFIX (Hauptprogramm).

Zusätzlich werden folgende COMPASS- bzw. MIX-FORTRAN-Unterprogramme verwendet:

MOVEC (siehe 3.4.01)

LOCATE.

Das Unterprogramm LOCATE wird aufgerufen mit  
CALL LOCATE(KANAL,IBLOCKNR,3).

Dabei ist KANAL die Kanalnummer einer Datei auf Magnetplatte und IBLOCKNR die Nummer des Blocks, auf den diese Datei positioniert werden soll.

OPEN (MIX-FORTRAN)

Das Unterprogramm OPEN wird aufgerufen durch

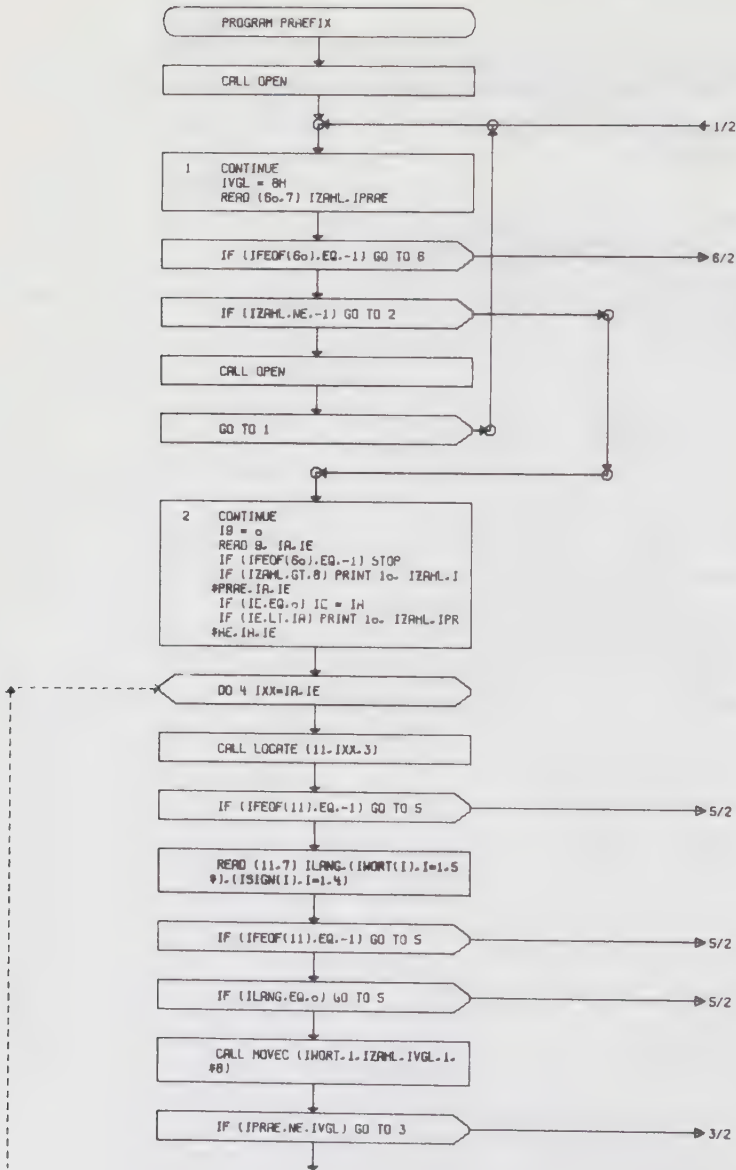
CALL OPEN

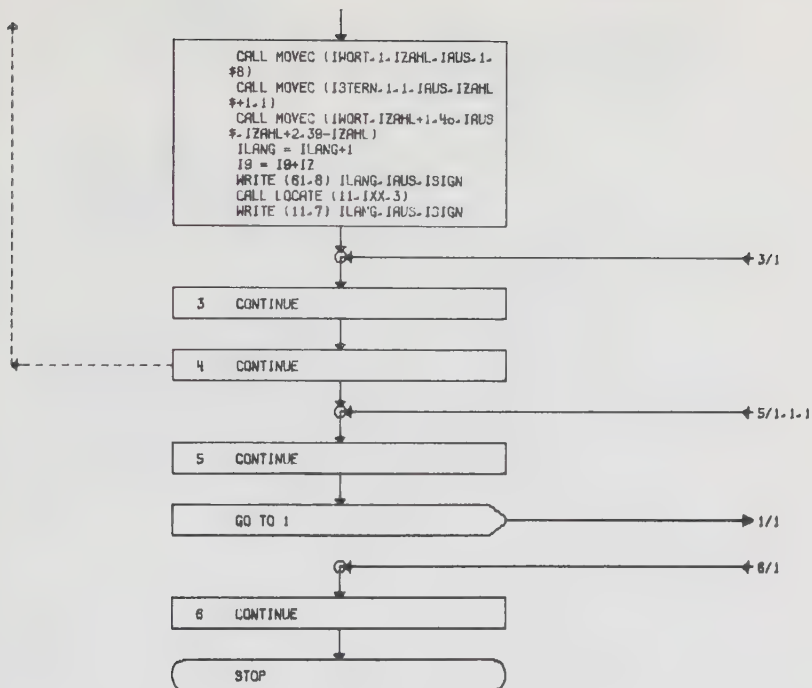
und öffnet die Datei, die die komprimierten Wortformen enthält, unter der Kanalnummer 11. Wenn schon eine Datei mit Kanalnummer 11 existiert, so wird diese erst geschlossen. Der Name der zu öffnenden Datei wird von der Standardeingabedatei (vom Kartenleser kommend) gelesen.

PROGRAM PRAEFIX

```
CHARACTER PRAE(8),PRFX(40),VGL(8),IFELD(40),IX(8)
DIMENSION IWORT(5), ISIGN(4)
EQUIVALENCE (IPRAE,PRAE(1)), (IWORT(1),PRFX(1),IFELD(1)), (IVGL,
$VGL(1)), (ISIGN(3),I2)
DIMENSION IAUS(5)
DATA ISTERN/1H*/
```

```
7  FORMAT (I2,1X,5A8,1X,2A8,1X,I9,I10)
8  FORMAT (1H ,I2,1H ,5A8,1X,2A8,1X,I9,I10,55X)
9  FORMAT (2I5)
10 FORMAT (10X,*****#,I2,1X,A8,2I6,5X,*FEHLER..IE.LT.IA.OR.
$IZAHL.GT.8*)
```





#### 3.4.06 Programmbeschreibung KETT

Das Programm KETT enthält nur eine FORTRAN-  
Programm

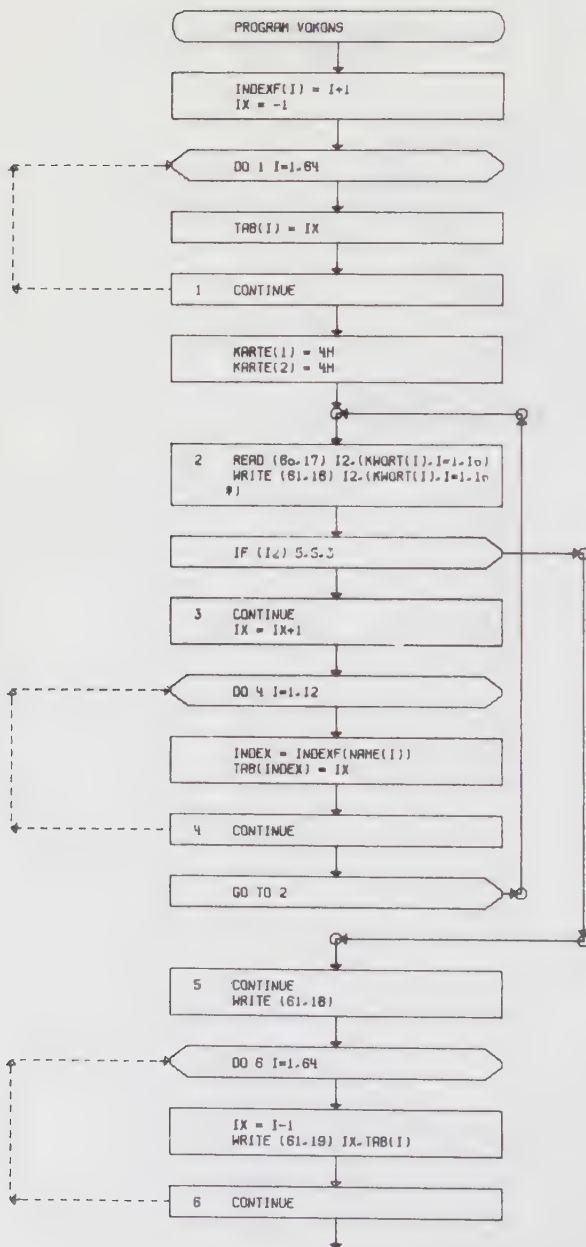
VOKONS (Hauptprogramm).

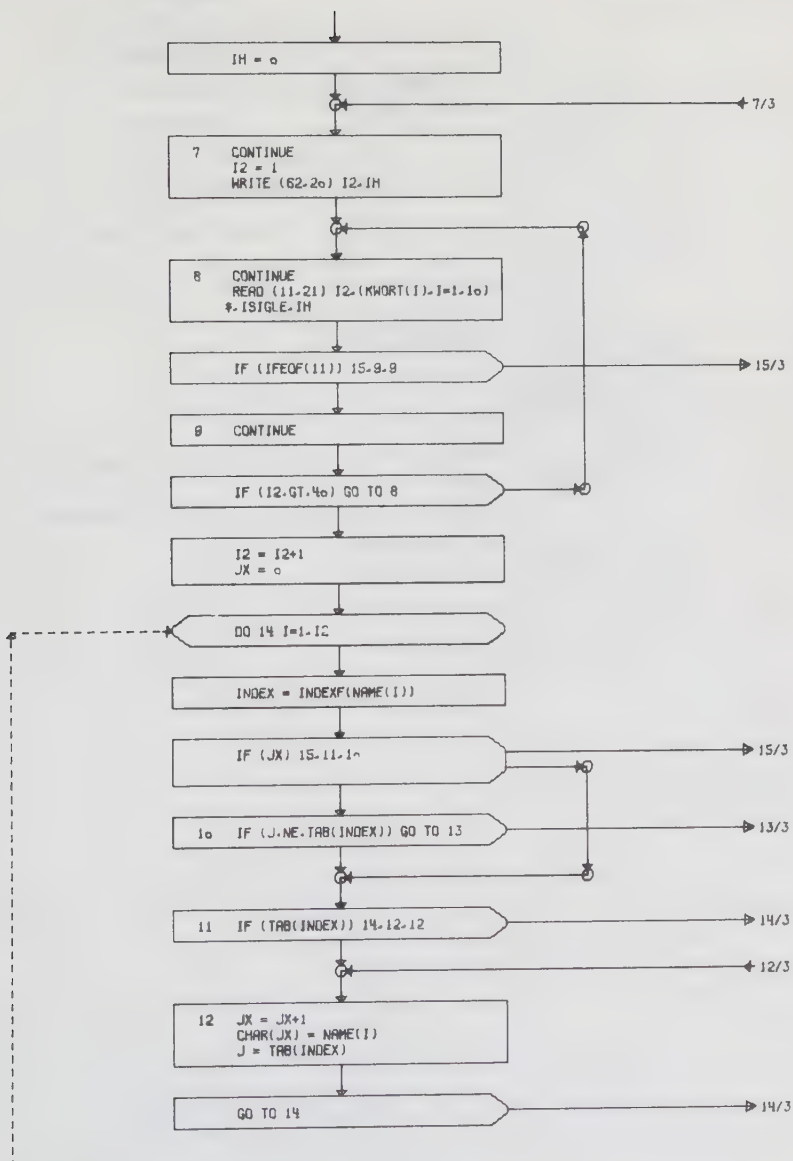
PROGRAM VOKONS

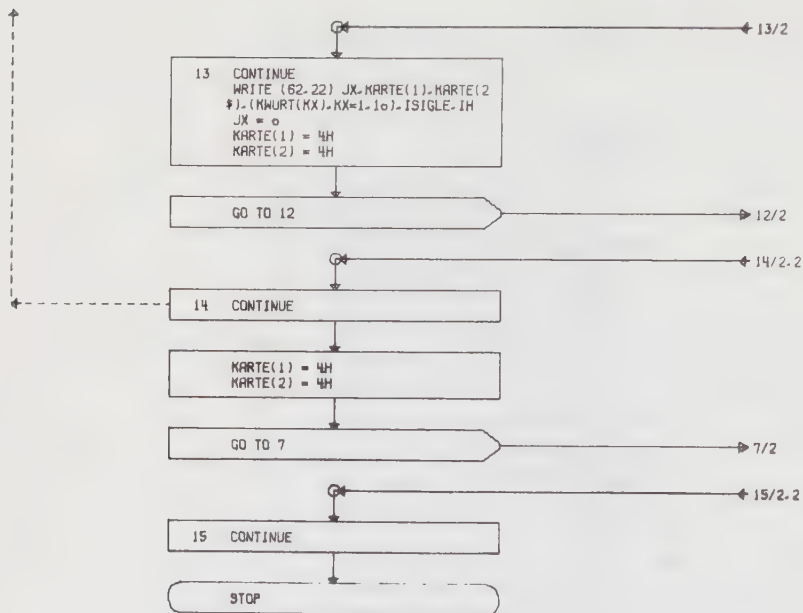
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

REAL ISIGLE  
CHARACTER NAME(40),CHAR(8)  
INTEGER TAB(64)  
DIMENSION KWORT(10), KARTE(2)  
EQUIVALENCE (NAME(1),KWORT(1)), (KARTE(1),CHAR(1))

16 FORMAT (1H ,5X,I2,10A4)  
17 FORMAT (I2,1X,10A4)  
18 FORMAT (1H1,/1HQ)  
19 FORMAT (1H ,R1,10X,I3)  
20 FORMAT (I2,61X,I7)  
21 FORMAT (I2,1X,10A4,1X,A8,3X,I9)  
22 FORMAT (I2,1X,2A4,2X,10A4,A8,2X,I7)







#### 3.4.o7 Programmbeschreibung KMPK

Das Programm KMPK besteht aus folgenden Teilprogrammen:

KETTKMPR (Hauptprogramm)

DATAKMPR

VGLSTR (siehe 3.4.o3)

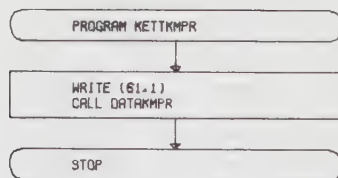
KOMPR (siehe 3.4.o3).

PROGRAM KETTKMPR

1     FORMAT (1H1,/1HR)

PROGRAM KETTKMPR

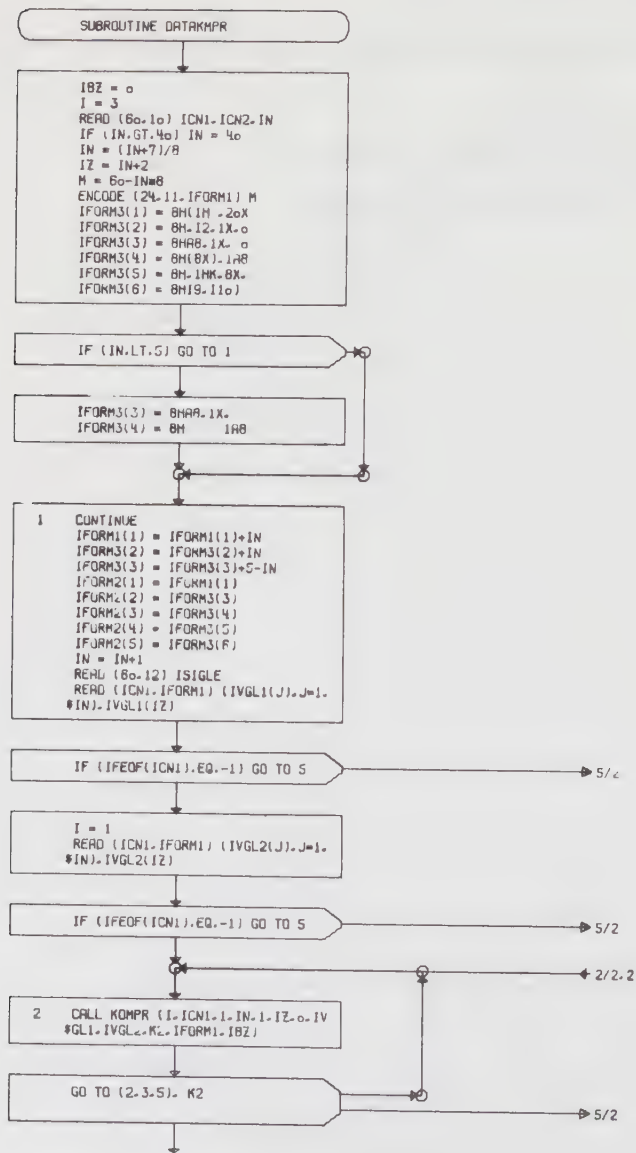
1

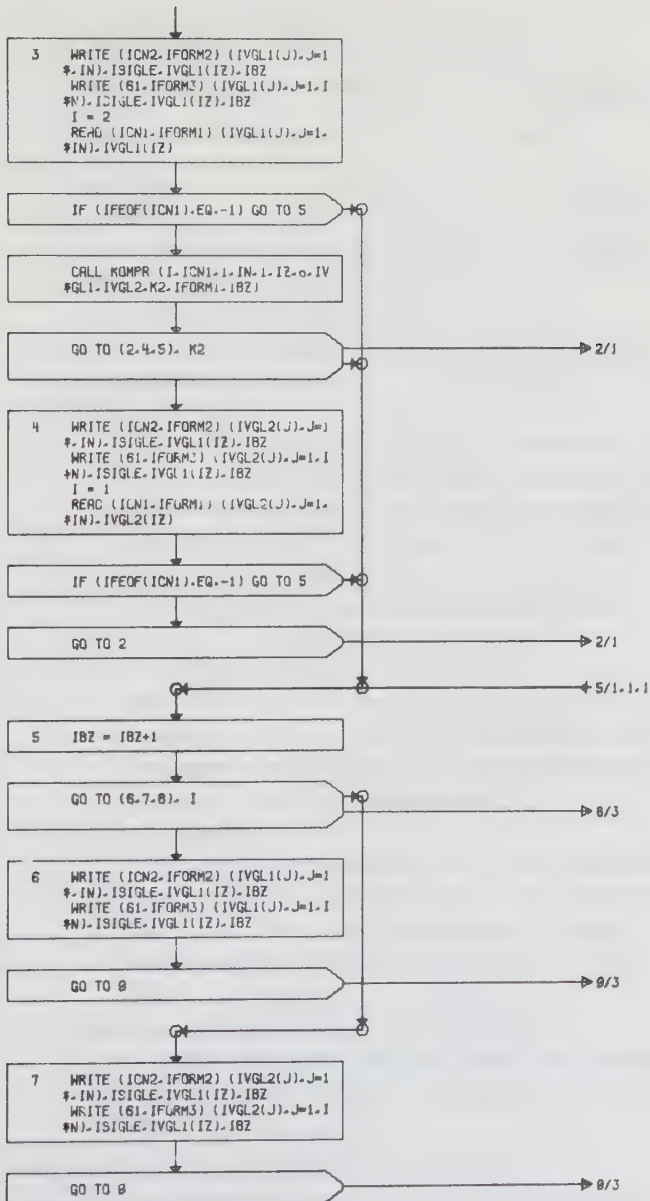


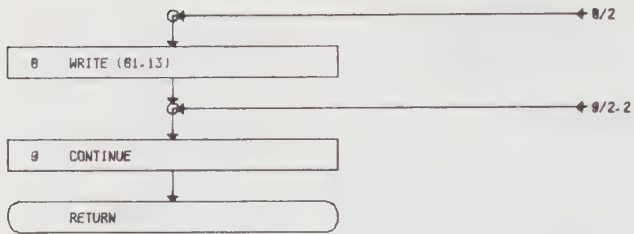
SUBROUTINE DATAKMPR

DIMENSION IVGL1(7), IVGL2(7), IFORM1(3), IFORM2(5), IFORM3(6)

```
10  FORMAT (3(I2,1X))
11  FORMAT (8H(I2,1X,0,3HA8,,I2,5HX,I7),6H      )
12  FORMAT (1A8)
13  FORMAT (20X,18HKEINE EINGABEDATEN)
```







### 3.4.08 Programmbeschreibung UEMK

Das Programm UEMK besteht aus folgenden Teilprogrammen:

STATIST (Hauptprogramm)  
TAB  
INDEXF  
KETTSTAT  
KINDEX  
DATAKMPR (siehe 3.4.07)  
VGLSTR (siehe 3.4.03)  
KOMPR (siehe 3.4.03)

Außerdem werden folgende FORTRAN- bzw. COMPASS-Unterprogramme verwendet:

XFER

Das Unterprogramm XFER ist ein FORTRAN-Programm, das die unter dem Betriebssystem MASTER zur Verfügung stehende Systemroutine gleichen Namens<sup>1</sup> zum Kopieren von Dateien ruft. Der Aufruf lautet:

```
CALL XFER(KANALIN,NWRDSIN,KANALOUT,  
          NWRDSOUT,O,O,O,O).
```

Dabei ist KANALIN die Kanalnummer und NWRDSIN die Zahl der Maschinenworte (zu je 4 Zeichen) pro Block zu kopierenden Datei, KANALOUT und NWRDSOUT die entsprechenden Parameter für die Ausgabedatei.

IBCDSI (COMPASS)

Das Funktionsunterprogramm IBCDSI verwandelt eine binäre Kanalnummer in ihr BCD-Äquivalent (vier Zeichen linksbündig). Dieses Unterprogramm ist eine Hilfsroutine für XFER.

Q9QCALL (Laufzeitroutine von FORTRAN)

Das Unterprogramm Q9QCALL wird hier verwendet um das Systemsortierprogramm SORT aufzurufen.

---

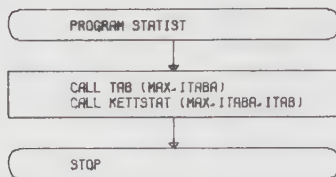
<sup>1</sup> Siehe: 3300 3500 Computer System MASTER Version 3 Reference Manual, Abschnitt 11.

PROGRAM STATIST

DIMENSION ITABA(2,64), ITAB(300)

PROGRAM STATIST

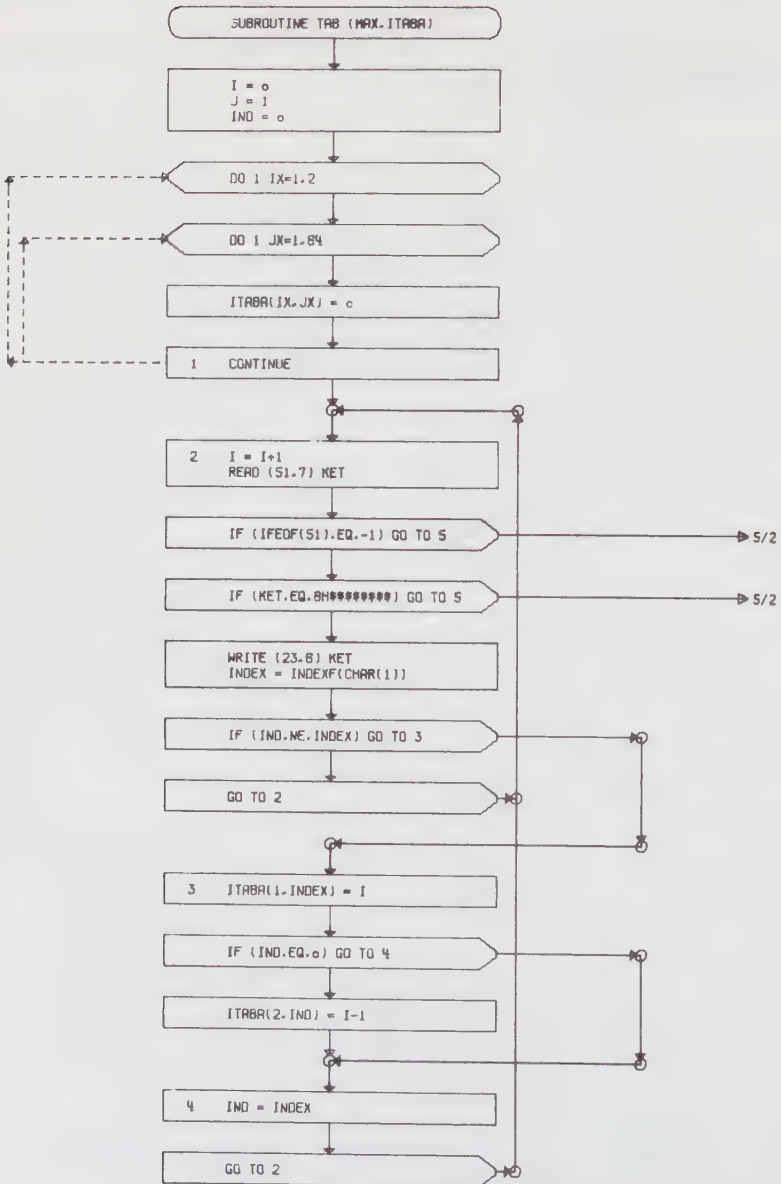
1

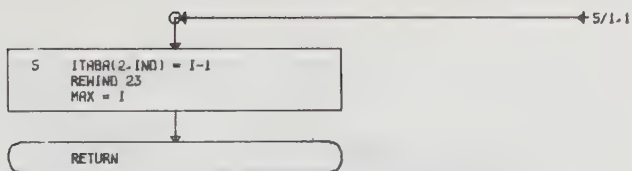


SUBROUTINE TAB (MAX,ITABA)

DIMENSION ITABA(2,64)  
CHARACTER CHAR(8)  
EQUIVALENCE (CHAR(1),KET)

6     FORMAT (A8)  
7     FORMAT (3X,A8)

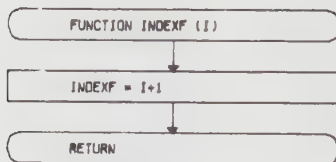




FUNCTION INDEXF (I)

FUNCTION INDEXF

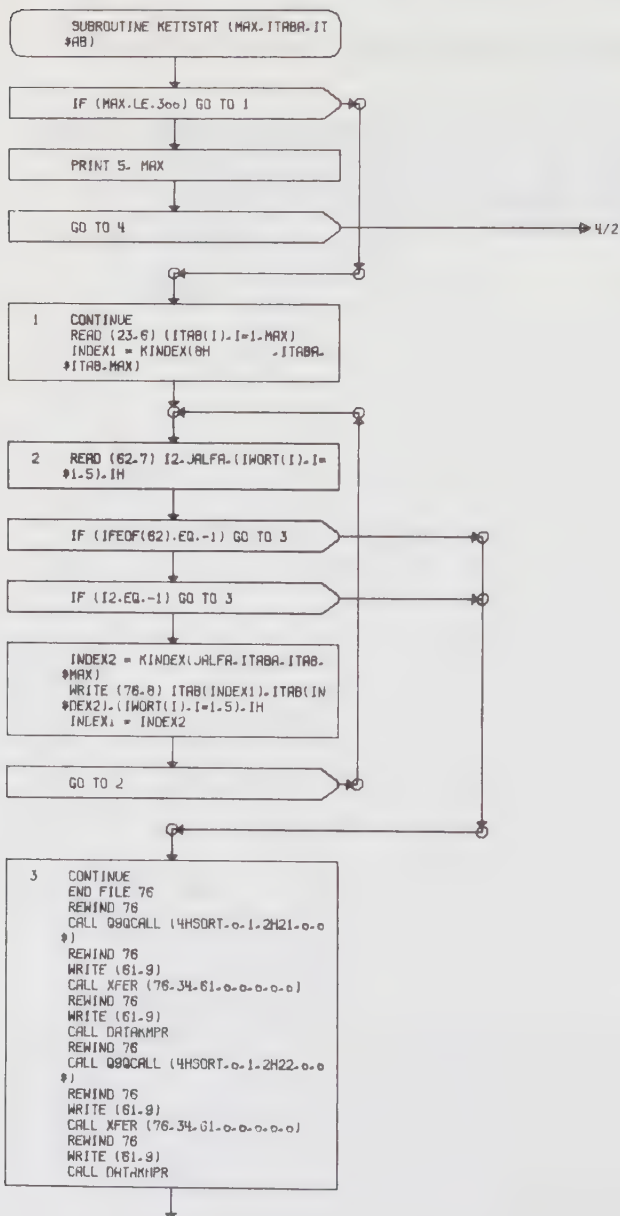
1



SUBROUTINE KETTSTAT (MAX,ITABA,ITAB)

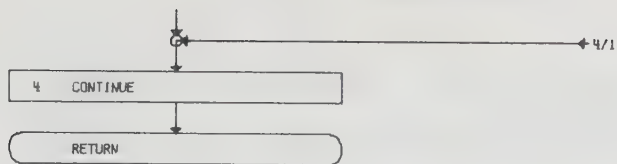
DIMENSION ITAB(MAX), ITABA(2,64), IWORT(5)

```
5  FORMAT (5X,8HZU VIELE,8HZEICHENK,8HETTEN      ,8HMAX =      ,I7)
6  FORMAT (A8)
7  FORMAT (I2,1X,A8,2X,5A8,10X,I7)
8  FORMAT (1H ,A8,4X,A8,5X,5A8,7X,I7,56X)
9  FORMAT (1HR,/1H1)
```



SUBROUTINE KETTSTAT

2



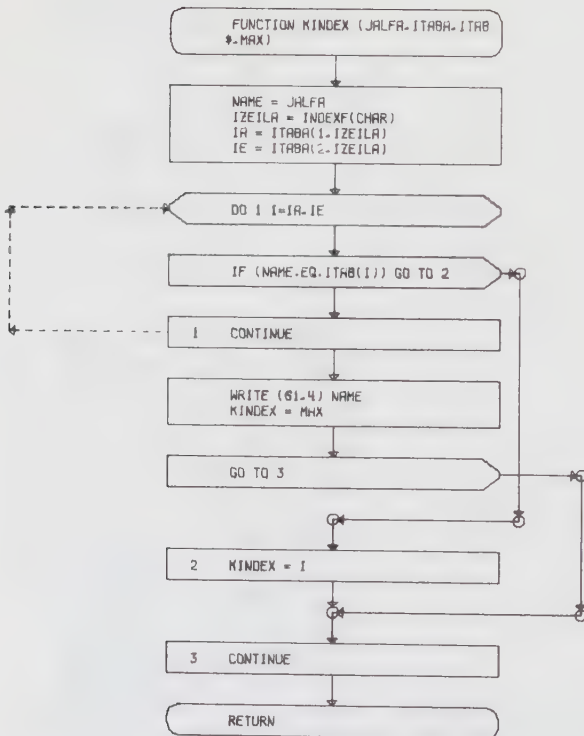
FUNCTION KINDEX (JALFA,ITABA,ITAB,MAX)

DIMENSION ITABA(2,64), ITAB(1)  
CHARACTER CHAR  
EQUIVALENCE (CHAR,NAME)

4      FORMAT (1H ,3HZEICHENK,3HETTE                      ,A3,2X,8HNICHT GE,8HFUNDEN )

FUNCTION KINDEX

1



### 3.4.09 Programmbeschreibung GRUL

Das Programm GRUL besteht aus folgenden Teilprogrammen:

GENERATE (Hauptprogramm)

GRULE.

Zusätzlich werden folgende FORTRAN- bzw. COMPASS-Unterprogramme verwendet:

SORTCTRL

Das FORTRAN-Unterprogramm SORTCTRL wird aufgerufen durch

CALL SORTCTRL(KANAL).

Es erzeugt aus einer Minimalinformation, die es von der Standard-Eingabedatei liest, Steuerinformationen für das unter MASTER verfügbare Standard-Sortierprogramm und schreibt sie auf die durch KANAL spezifizierte Datei. Gegebenenfalls werden Dateien zum Zwischenspeichern beim Sortieren zur Verfügung gestellt.

ISCAN (siehe 3.4.01)

ICMPR (siehe 3.4.04)

MOVEC (siehe 3.4.01).

Nach diesem Programm wird das Standardprogramm SORT<sup>1</sup> aufgerufen, um die Regeldatei zu sortieren.

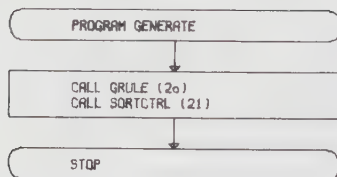
---

<sup>1</sup>Vgl. 3.4.02, Anm.1.

# PROGRAM GENERATE

PROGRAM GENERATE

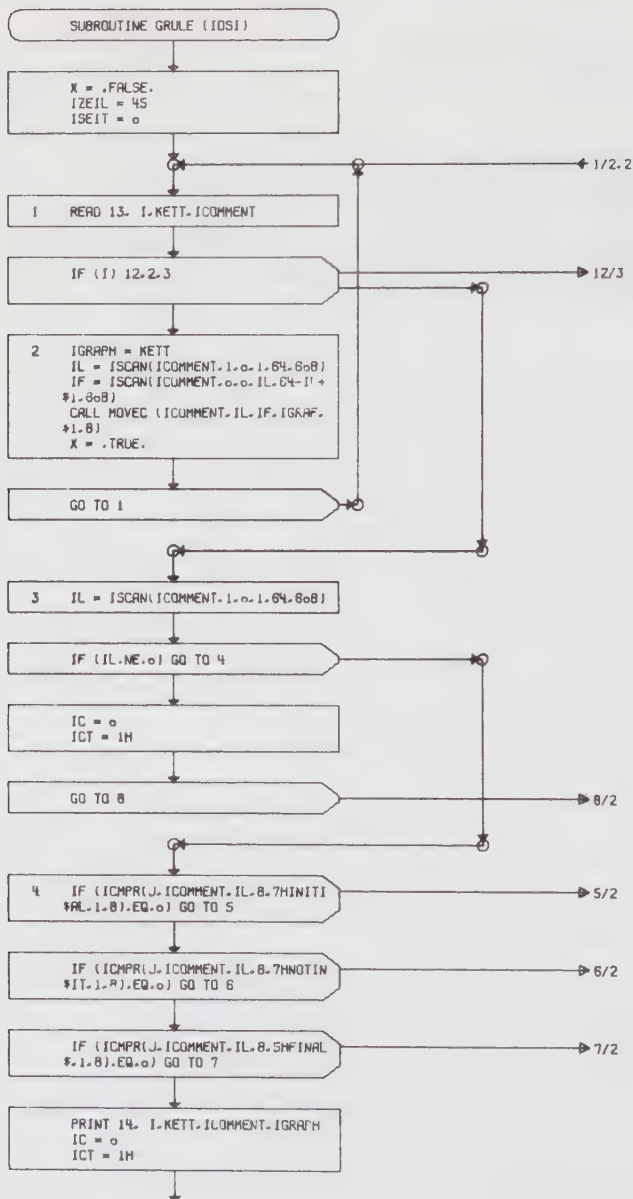
1

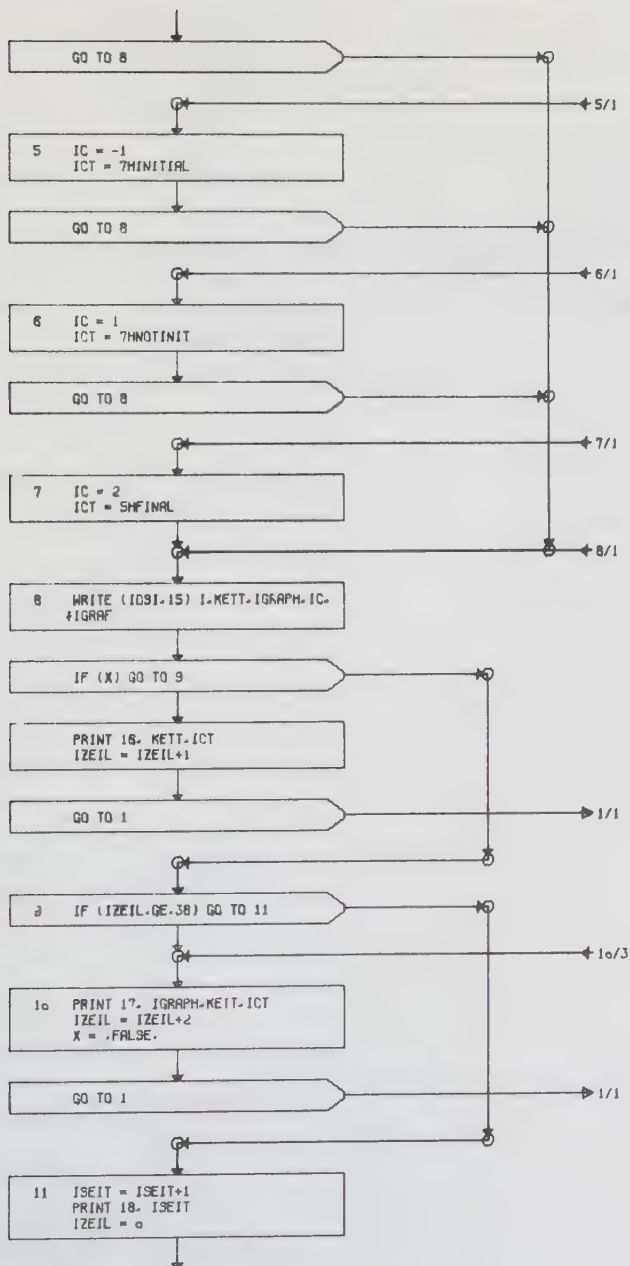


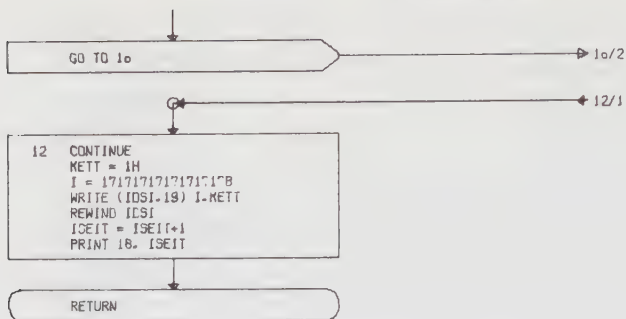
SUBROUTINE GRULE (IDSI)

DIMENSION ICOMMENT(8)  
LOGICAL X

```
13  FORMAT (I2,1X,9A8)
14  FORMAT (1H ,8HFEHLER  ,I2,1X,9A8,5X,A8)
15  FORMAT (I2,2A8,I2,A8)
16  FORMAT (1H ,48X,2A8)
17  FORMAT (1H-,10X,7HGRAPHEN,3X,A8,8X,9HVARIANTEN,3X,2A8)
18  FORMAT (1H1,62X,6HSEITE ,I2/1H )
19  FORMAT (A2,A8)
```







#### 3.4.10 Programmbeschreibung TRNS

Das Programm TRNS besteht aus folgenden Teilprogrammen:

TRANS (Hauptprogramm)

TRANSLAT

TAB

INDEXF (siehe 3.1.10.1)

GETRULE.

Zusätzlich werden die folgenden COMPASS-Unterprogramme verwendet:

ISCAN (siehe 3.4.01)

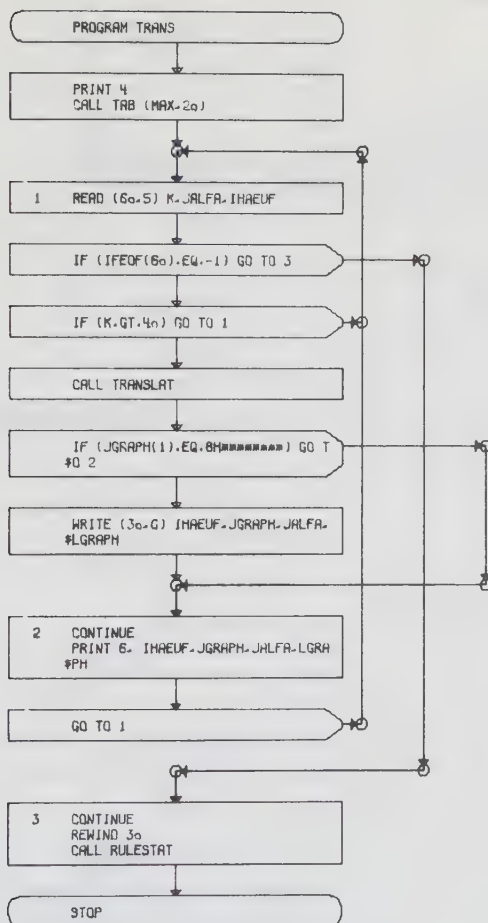
ICMPR (siehe 3.4.04)

MOVEC (siehe 3.4.01).

PROGRAM TRANS

```
COMMON /WORTFRM/ JALFA(5),JGRAPH(5),LGRAPH(5)
COMMON /HAEUF/ IHAEUF
COMMON /RULEDIR/ ITABA(2,64),IRULE(4,500),IA,ITAB(2),IC,IGRAF
COMMON /MAX/ MAX
```

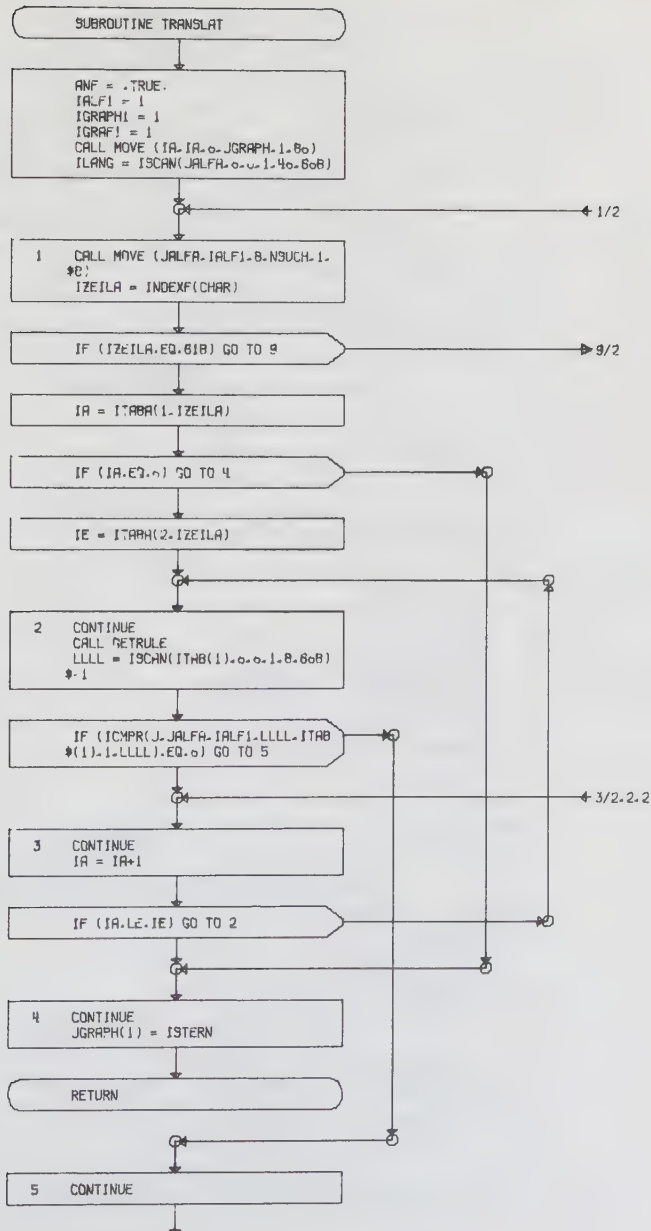
```
4  FORMAT (1HR)
5  FORMAT (I2,1X,5A8,20X,I7)
6  FORMAT (1X,I7,2X,5A8,3X,5A8,3X,5A8)
```

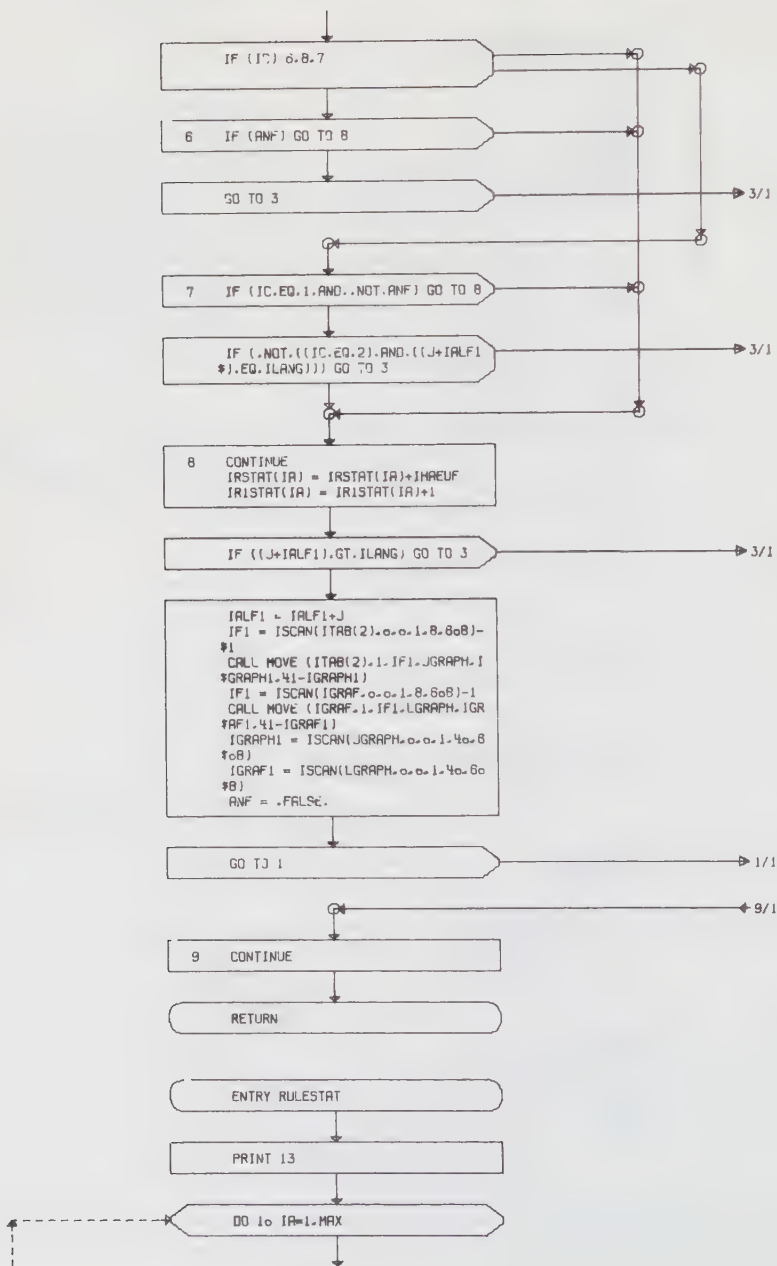


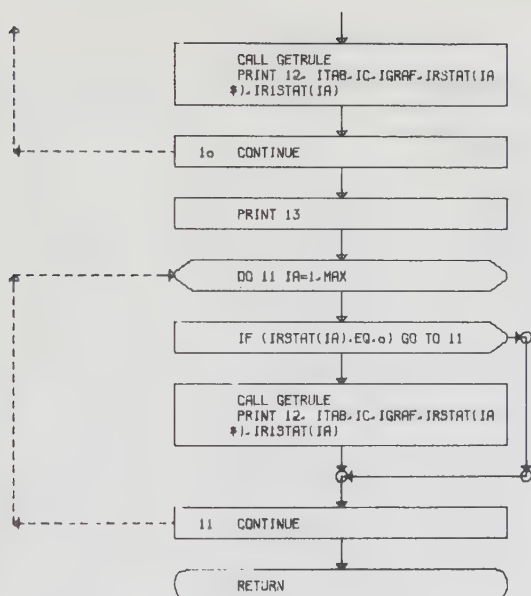
SUBROUTINE TRANSLAT  
ENTRY RULESTAT

DIMENSION IRSTAT(500)  
DIMENSION IR1STAT(500)  
COMMON /HAEUF/ IHAEUF  
COMMON /RULEDIR/ ITABA(2,64),IRULE(4,500),IA,ITAB(2),IC,IGRAF  
COMMON /WORDFORM/ JALFA(5),JGRAPH(5),LGRAPH(5)  
COMMON /MAX/ MAX  
LOGICAL ANF  
CHARACTER CHAR  
EQUIVALENCE (CHAR,NSUCH)  
DATA IRSTAT/500\*0/  
DATA IR1STAT/500\*0/  
DATA ISTERN/8H\*\*\*\*\*/

- 12   FORMAT (2X,2A8,I2,A8,5X,I9,23H = ZAHL DER ANWENDUNGEN,5X,\*(\*,I9,\*)  
     \$\*)  
13   FORMAT (1H1)



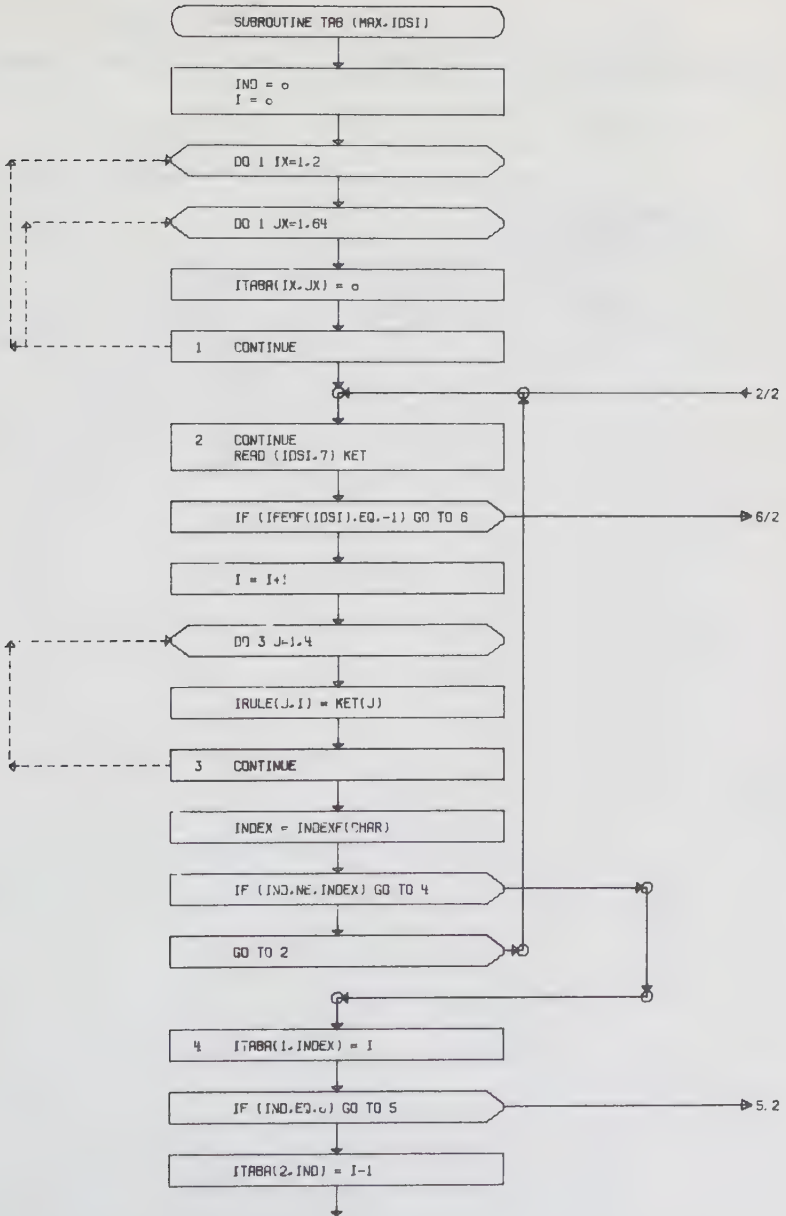


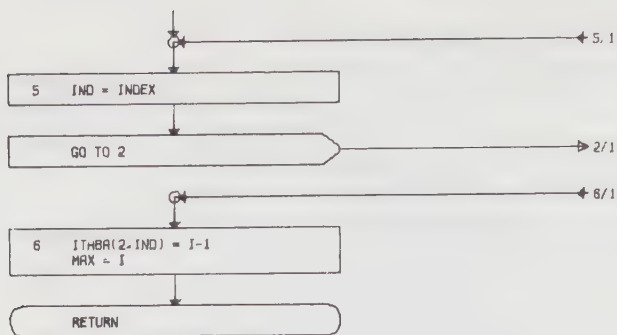


SUBROUTINE TAB (MAX, IDSI)

COMMON /RULEDIR/ ITABA(2,64),IRULE(4,500),IA,ITAB(2),IC,IGRAF  
CHARACTER CHAR  
DIMENSION KET(4)  
EQUIVALENCE (CHAR,KET(1))

7     FORMAT (2X,2A8,I2,A8)



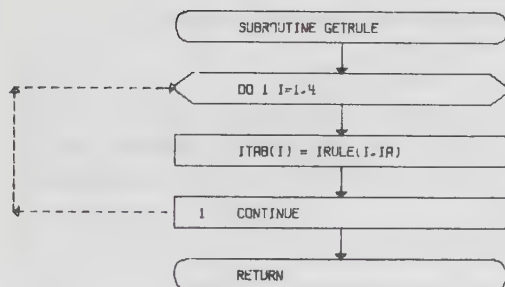


SUBROUTINE GETRULE

COMMON /RULEDIR/ ITAB(2,64),IRULE(4,500),IA,ITAB(2),IC,IGRAF

SUBROUTINE GETRULE

1



### 3.4.11 Programmbeschreibung UEMG

Das Programm UEMG besteht aus folgenden Teilprogrammen:

GRAFSTAT (Hauptprogramm)

SIGMA

RRULEGEN

MASSFKTN

AUSGABE

EUKLDIFF

Zusätzlich wird folgendes COMPASS-Unterprogramm verwendet:

RELMEM

Das Unterprogramm RELMEM gibt allen nicht verwendeten Kernspeicherplatz an das Betriebssystem zurück. Es wird aufgerufen durch

CALL RELMEM.

PROGRAM GRAFSTAT

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```

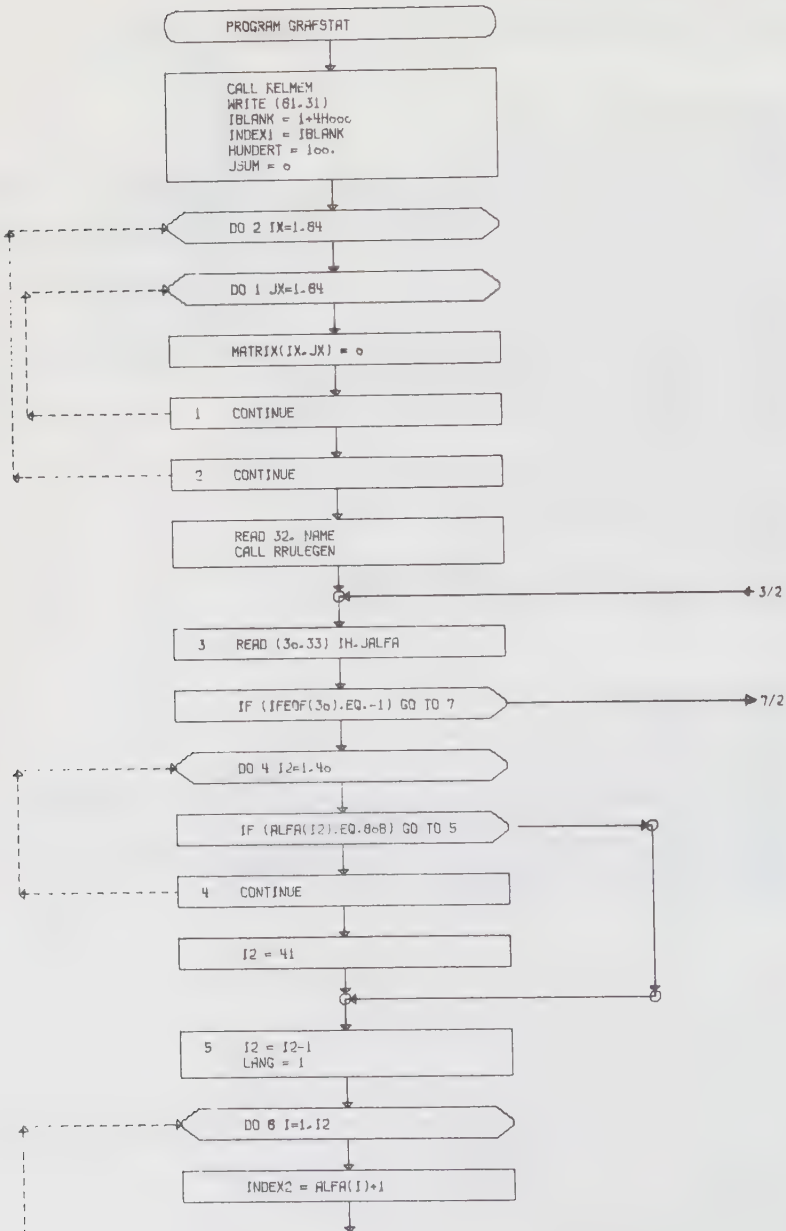
DIMENSION NAME(10)
COMMON /PAR/ IX,JX,IENDO
COMMON /SCHW/ ISCHW,IDIF(40)
DIMENSION IVMTRX(10,40), INMTRX(10,40), RMATRIX1(40,40), RMATRIX2
$(40,40), IREL1(8), IREL2(8), IRELABS1(10), IRELABS2(10)
DIMENSION RSUMPROZ(40), NSUM(64)
CHARACTER REL1(32),REL2(32),RELABS1(40),RELABS2(40),VORMTRX(40,40)
$,NACHMTRX(40,40)
EQUIVALENCE (REL1,IREL1), (REL2,IREL2), (RELABS1,IRELABS1),
$(RELABS2,IRELABS2), (IVMTRX,VORMTRX), (INMTRX,NACHMTRX)
COMMON /MASS/ MFAKT,SCHWEXP
COMMON /DATEN/ I,J,RMATRIX1,RMATRIX2,XX1,XX2,ALOGV,ALOGN,IREL1,
$IREL2,IRELABS1,IRELABS2,LI
COMMON /IRP/ IRPMAX,IRPOINT(64)
CHARACTER CHAR(4),ALFA(40)
DIMENSION JALFA(10)
DIMENSION MATRIX(64,64)
EQUIVALENCE (ALFA(1),JALFA(1))
COMMON /RRULE/ IRULE(64)
COMMON /RUL/ LRULE(40)
COMMON MATRIX
COMMON /MTX/ IVMTRX,INMTRX,RVOR,RNACH
DIMENSION LANGWORT(40), LANGWRTF(40)
COMMON /SIGMAN/ SIGMA1,SIGMA2,SIGMA11,SIGMA12

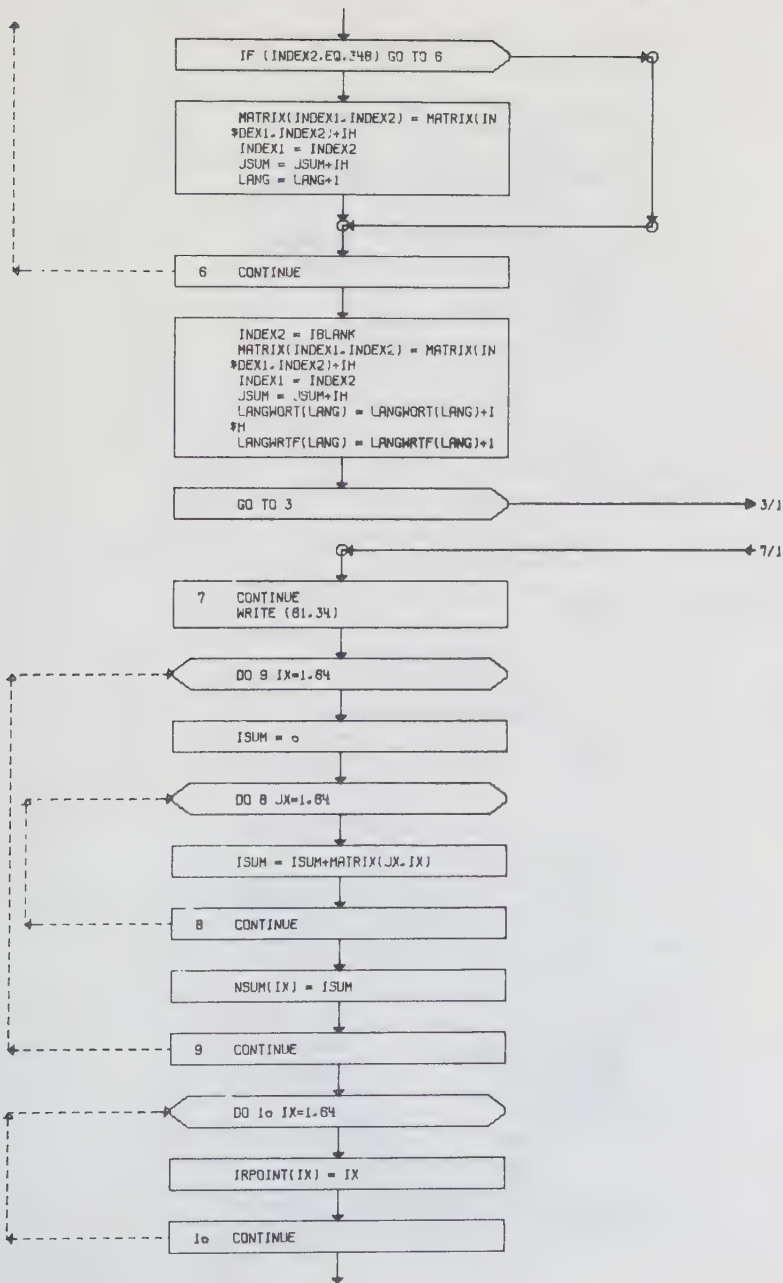
```

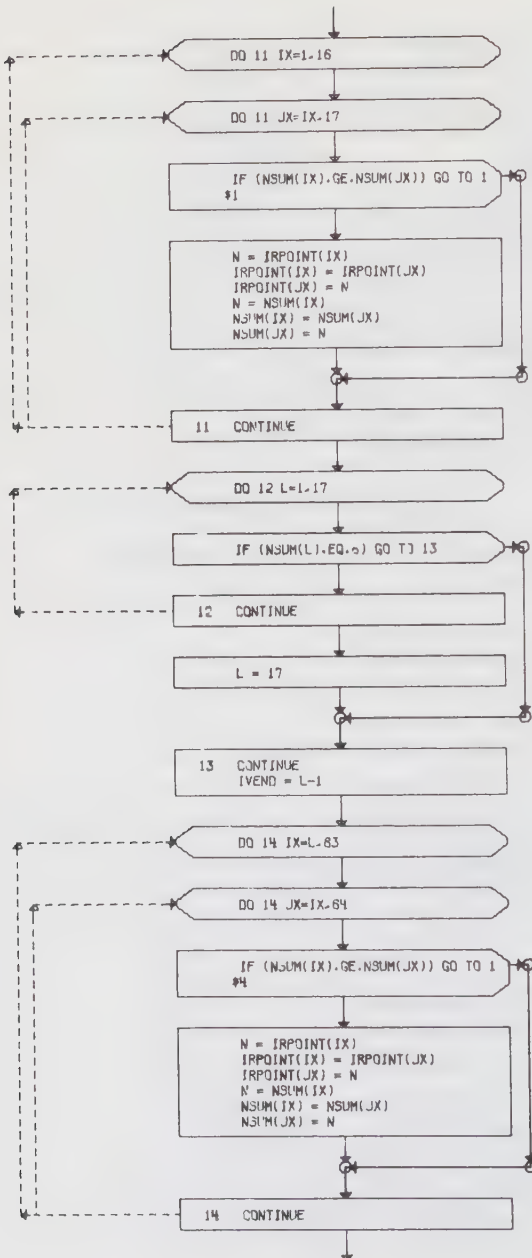
```

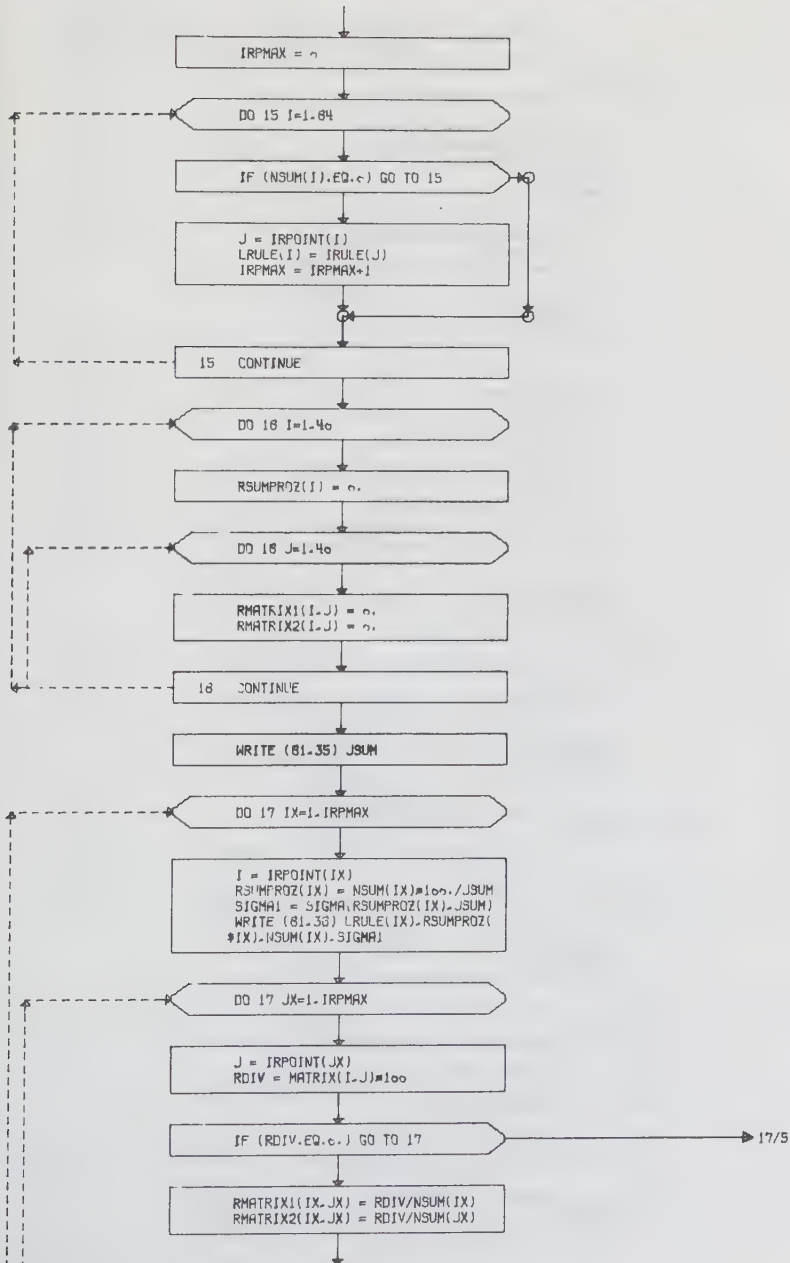
31  FORMAT (1H1,/1HR)
32  FORMAT (10A4)
33  FORMAT (1X,I7,38X,10A4)
34  FORMAT (1H1,/1HQ)
35  FORMAT (*1*,10X,*ZAHl ALLER GRAPHEME = *,I12/*0*)
36  FORMAT (10X,*GRAPHEM *,A4,5X,*REL. HAEUF. =*,F8.4,5X,*ABS.HAEUF. =
$,I12,5X,*STANDARDABWEICHUNG =*,F8.4)
37  FORMAT(*1*/5X,*LAENGE  ZAHl DER LAUFENDEN WORTFORMEN  ZAHl DER
$UNTERSCHIEDL. WORTFORMEN*/(6X,I2,10X,I12,20X,I12))
38  FORMAT (360(4X,9A8/),(4X,19A4))

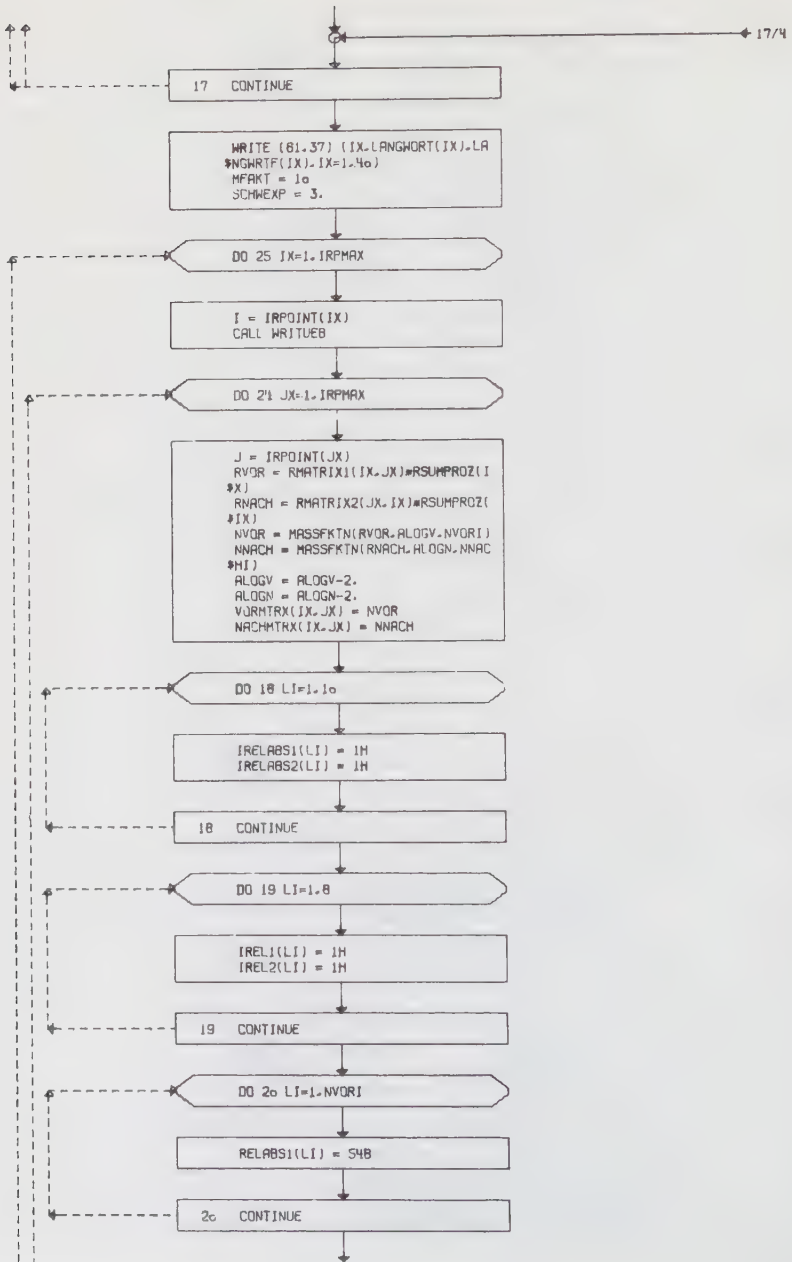
```

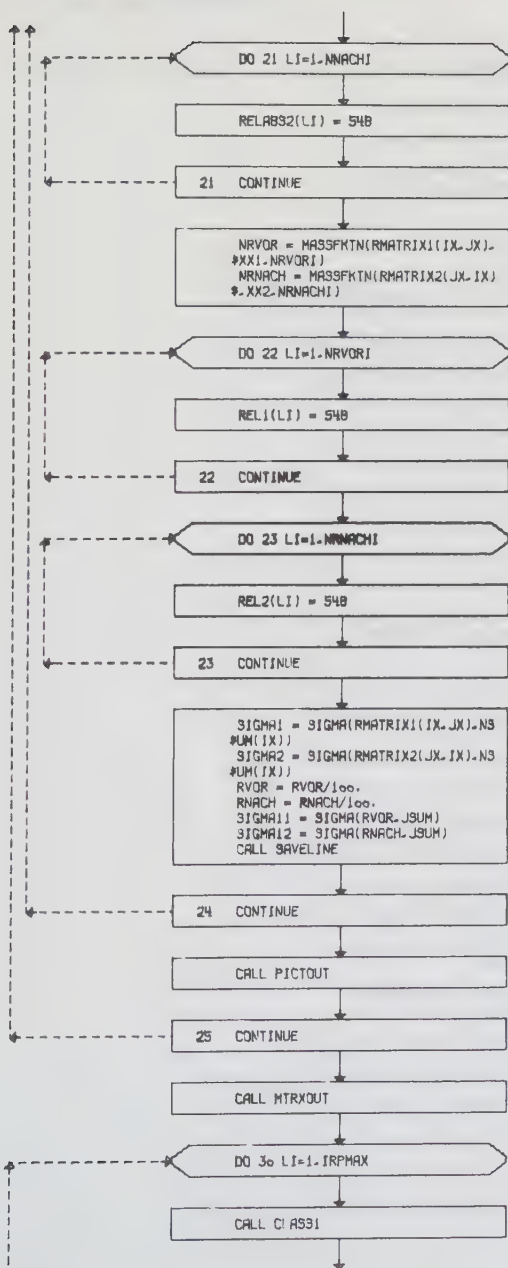


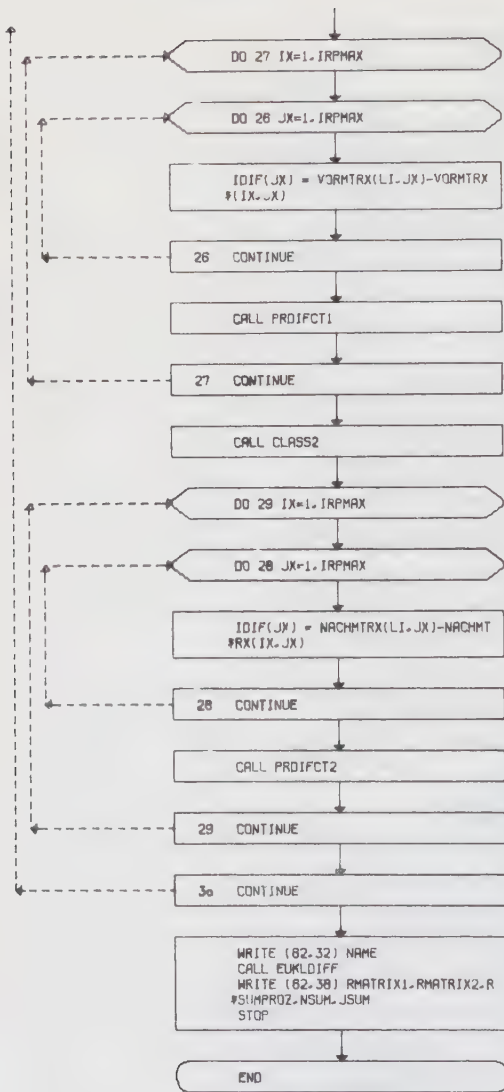










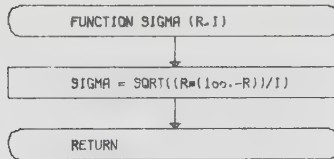


FUNCTION SIGMA (R,I)

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

FUNCTION SIGMA

1



SUBROUTINE RRULEGEN

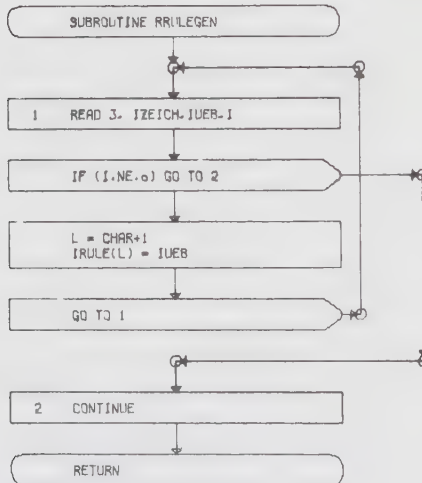
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /RRULE/ IRULE(64)  
CHARACTER CHAR  
EQUIVALENCE (CHAR,IZEICH)

3     FORMAT (A1,A4,4X,I2)

SUBROUTINE RRULEGEN

1



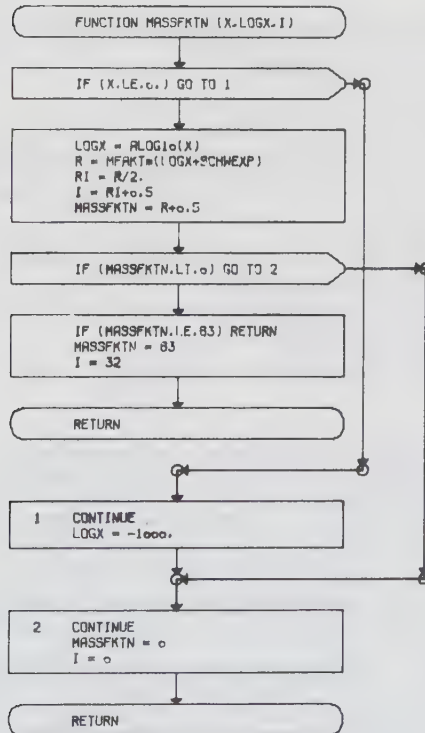
FUNCTION MASSFKTN (X,LOGX,I)

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /MASS/ MFAKT,SCHWEXP  
REAL LOGX

FUNCTION MASSFKTN

1



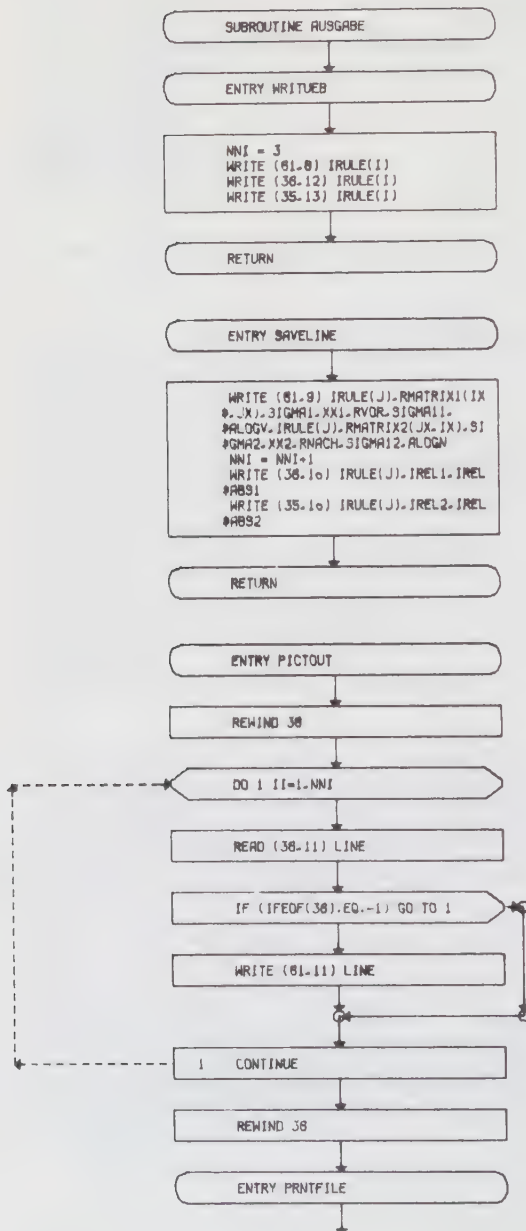
SUBROUTINE AUSGABE  
ENTRY WRITUEB  
ENTRY SAVEDLINE  
ENTRY PICTOUT  
ENTRY PRNTFILE  
ENTRY MTRXOUT  
ENTRY CLASS1  
ENTRY CLASS2  
ENTRY PRODFCT1  
ENTRY PRODFCT2  
ENTRY PAGE35

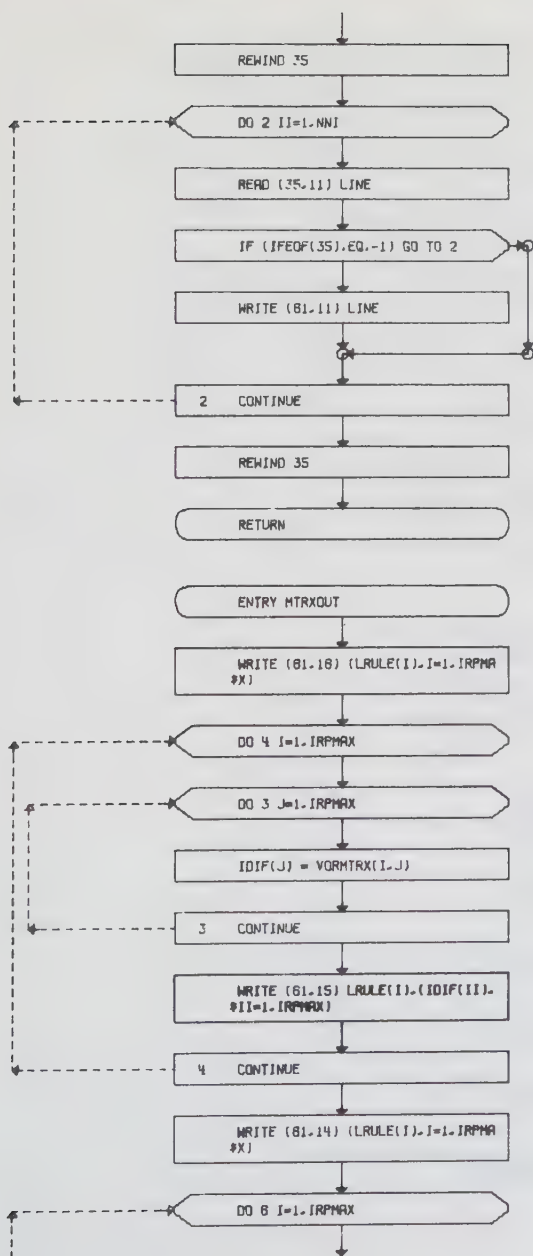
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

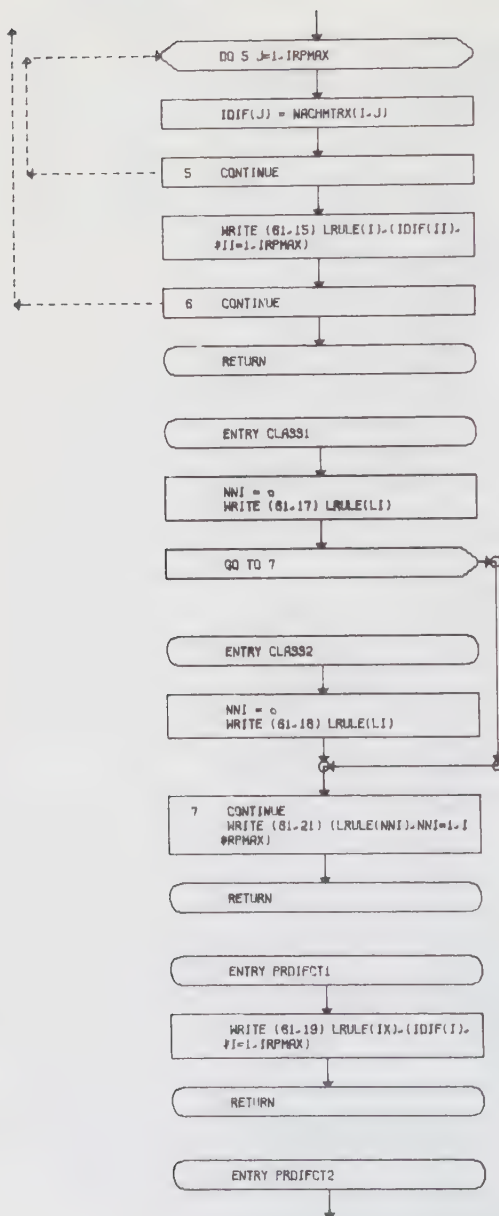
COMMON /SIGMAN/ SIGMA1,SIGMA2,SIGMA11,SIGMA12  
COMMON /IRP/ IRPMA,IRPOINT(64)  
COMMON /RRULE/ IRULE(64)  
DIMENSION LINE(34)  
COMMON /PAR/ IX,JX,IVEND  
COMMON /SCHW/ ISCHW,IOIF(40)  
DIMENSION IVMTRX(10,40), INMTRX(10,40), RMATRIX1(40,40), RMATRIX2  
\$(40,40), IREL1(8), IREL2(8), IRELABS1(10), IRELABS2(10)  
CHARACTER REL1(32),REL2(32),RELABS1(40),RELABS2(40),VORMTRX(40,40)  
\$,NACHMTRX(40,40)  
EQUIVALENCE (REL1,IREL1), (REL2,IREL2), (RELABS1,IRELABS1),  
\$(RELABS2,IRELABS2), (IVMTRX,VORMTRX), (INMTRX,NACHMTRX)  
COMMON /MASS/ MFAKT,SCHWEXP  
COMMON /DATEN/ I,J,RMATRIX1,RMATRIX2,XX1,XX2,ALOGV,ALOGN,IREL1,  
\$IREL2,IRELABS1,IRELABS2,LI  
COMMON /RUL/ LRULE(40)  
COMMON /MTX/ IVMTRX,INMTRX,RVOR,RNACH

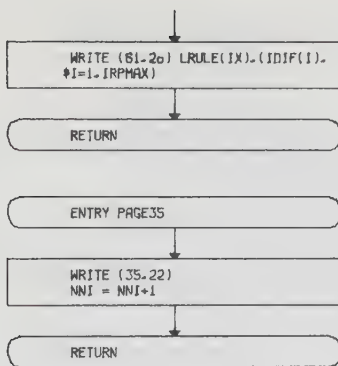
8     FORMAT(\*1\*,55X,\*GRAPHEM \*,A4/11X,\*ZAHL DES 1. GRAPHEMS = 100% ZAHL  
\$ ALLER GRAPHEME = 100%\*,3X,     \*ZAHL DES 2. GRAPHEMS = 100% ZAHL A  
\$LLER GRAPHEME = 100%\*/6X,\*VOR \*,2(4X,\*P     SIGMA(P) LOG10(P)\*,3X  
\$),1X,\*NACH \*,2(4X,\*P     SIGMA(P) LOG10(P)\*,3X)/5X,128(1H-))  
9     FORMAT (5X,2(1X,A4,1X,2(2(F8.4,1X),F8.5,3X)))  
10     FORMAT(\*     \*,A4,1X,\*I\*,8A4,\*I\*,10A4)  
11     FORMAT (34A4)  
12     FORMAT(\*1     NACHBEREICH DES GRAPHEMS \*,A4/11X,\*-3     -2     -1     0  
\$2     -5     -4     -3     -2     -1     0     1\*/12X,\*I\*,5(\*----I\*),7X,\*I\*,  
\$6(\*----I\*))  
13     FORMAT(\*1     VORBEREICH DES GRAPHEMS \*,A4/11X,\*-3     -2     -1     0  
\$1     2     -5     -4     -3     -2     -1     0     1\*/12X,\*I\*,5(\*----I\*),  
\$7X,\*I\*,6(\*----I\*))  
14     FORMAT(\*1VORBEREICH DES SPALTENGRAPHEMS\*/11X,40A3)  
15     FORMAT (5X,A3,3X,40I3)  
16     FORMAT(\*1NACHBEREICH DES SPALTENGRAPHEMS\*/11X,40A3)  
17     FORMAT(\*1\*20X,\* AEHNLICHKEIT \*,\* ZU GRAPHEM \*,A4.\*BEZUEGLICH DES  
\$\*NACHBEREICHES\*)

```
18  FORMAT(*1*,20X,* AEHNlichkeit *,* ZU GRAPHEM *,A4,*BEZUEGLICH DES
    $*VORBEREICHES*)
19  FORMAT (5X,A3,3X,40I3)
20  FORMAT (5X,A3,3X,40I3)
21  FORMAT (1X,135(1H*)/2X,*GRAPHEM*,2X,40A3/53(1H-))
22  FORMAT (*1*)
```







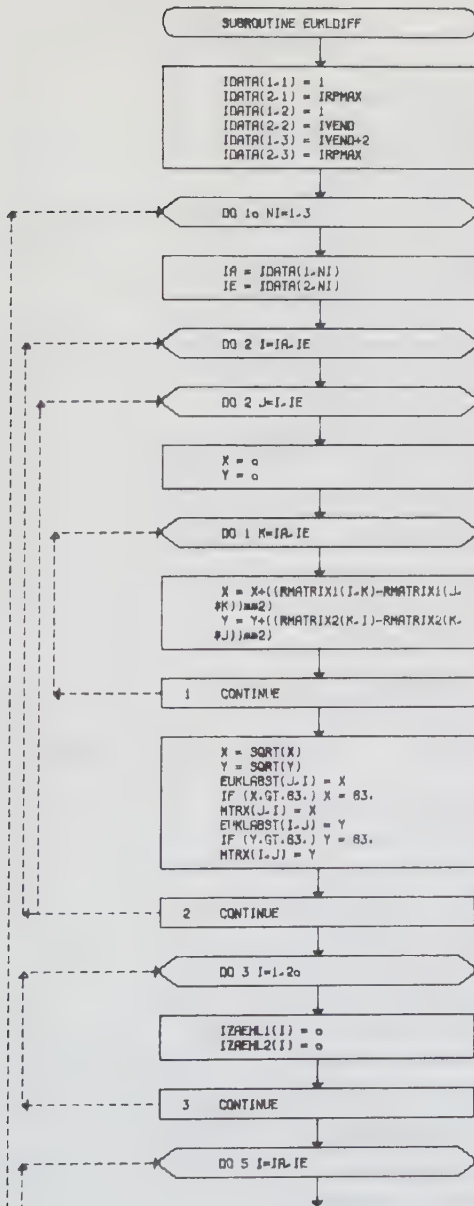


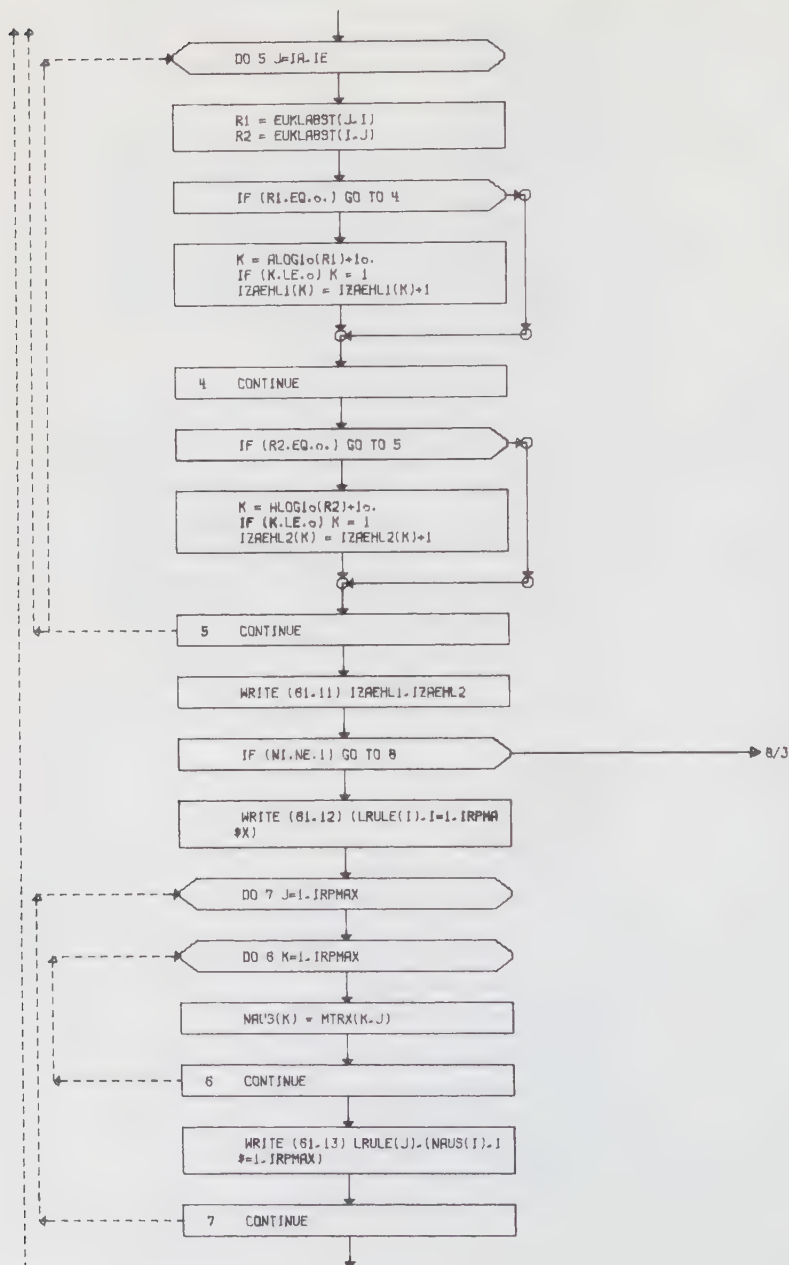
SUBROUTINE EUKLODIFF

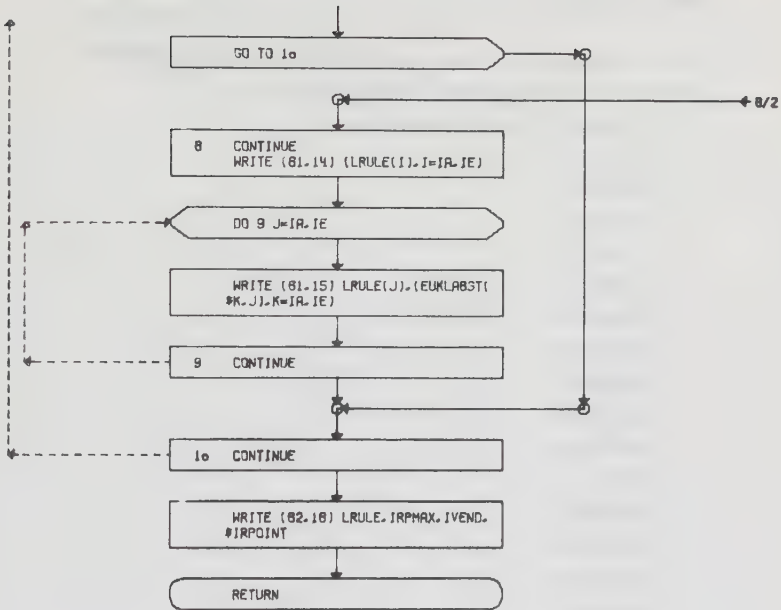
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON EUKLABST(40,40), IZAHL1(20), IZAHL2(20), R1, R2
COMMON /MTX/ IMTRX(10,40), NAUS(40)
CHARACTER MTRX(40,40)
EQUIVALENCE (IMTRX, MTRX)
COMMON /IRP/ IRPMAX, IRPOINT(64)
COMMON /RUL/ LRULE(40)
COMMON /DATEN/ I, J, RMATRIX1(40,40), RMATRIX2(40,40)
COMMON /PAR/ IX, JX, IVEND
DIMENSION IDATA(2,3)
```

```
11  FORMAT(*1*,14X,*.LE.E-9 E-8 E-7 E-6 E-5 E-4 E-3 E-2
    $E-1 E0 E+1 E+2 E+3 E+4 E+5 E+6 E+7 E+8 E+9
    $E+10*/4X,*VORBEREICH *,20I6/4X,*NACHBEREICH*,20I6)
12  FORMAT(*1 RECHTE OBERE HAELFTE VORBEREICH, LINKE UNTERE HAELFTE
    $NACHBEREICH*/10X,40A3)
13  FORMAT (2X,A4,3X,40I3)
14  FORMAT(*1 RECHTE OBERE HAELFTE VORBEREICH, LINKE UNTERE HAELFTE
    $NACHBEREICH*/10X,25(A4,1X))
15  FORMAT (3X,A4,2X,25F5.1)
16  FORMAT (20A4/20A4/2I12/40I2/24I2)
```







### 3.4.12 Programmbeschreibung für die Operatoren KONX und CLSS

Das Programm für die Operatoren KONX und CLSS besteht aus folgenden Teilprogrammen:

CLASSIFY (Hauptprogramm)  
INPUT  
NEWPAGE  
CLRLINE  
OUTPUT  
RSORT  
GARBAGE  
MATRIXOP  
NTUPLE  
TSTTUPEL  
GETTUPEL  
STOP  
SCHWELL  
EUKLID  
EUKCLASS  
CONNECT

Zusätzlich werden folgende COMPASS-Unterprogramme verwendet:

INLIST

Das Unterprogramm INLIST wird aufgerufen durch

CALL INLIST(INDEX,ELEMENT).

Dadurch wird in ein COMMON 2 CHARACTER-Feld in das Feldelement mit dem Index INDEX das Zeichen ELEMENT eingetragen.

CLIST

Das Unterprogramm CLIST ist unter FORTRAN als INTEGER FUNCTION aufzurufen durch

I=CLIST(INDEX),

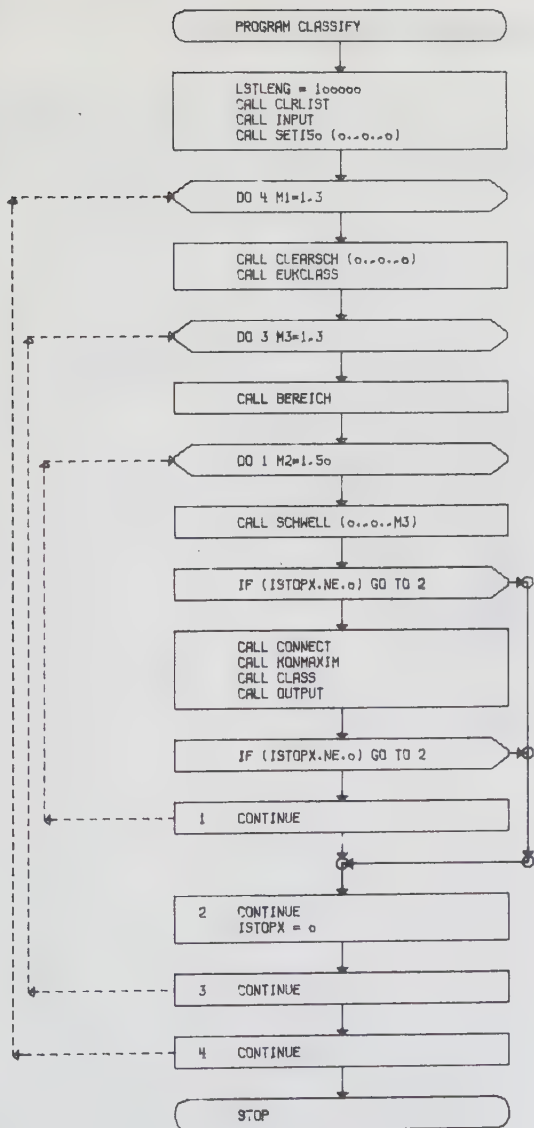
wenn der Name CLIST vorher als INTEGER vereinbart wurde. Dieses Unterprogramm ist das Gegenstück zu INLIST. Nach dem Aufruf hat

die Variable I den Wert des Zeichens des  
Feldelements mit dem Index INDEX. INLIST  
und CLIST beziehen sich auf den gleichen  
Adressbereich im COMMON 2.

PROGRAM CLASSIFY

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /LST/ LISTHEAD,LSTLENG  
COMMON /STOPX/ ISTOPX



# SUBROUTINE INPUT

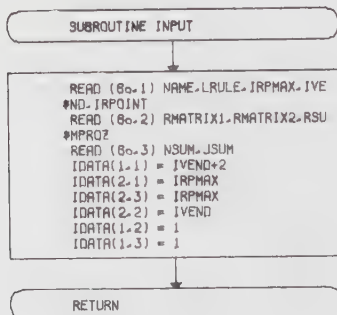
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON /X/ NAME(10),LRULE(40),IRPMAX,IVEND,IRPOINT(64),EUKLABST(40  
$,40),RMATRIX1(40,40),RMATRIX2(40,40),RSMUPROZ(40),NSUM(64),JSUM  
COMMON /DATA/ IDATA(2,3)
```

```
1  FORMAT (10A4/2(20A4/),2I12/40I2/24I2)  
2  FORMAT (4X,9A8)  
3  FORMAT (4X,19A4)
```

SUBROUTINE INPUT

1



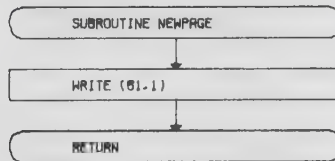
SUBROUTINE NEWPAGE

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

1     FORMAT (1H1)

SUBROUTINE NEWPAGE

1



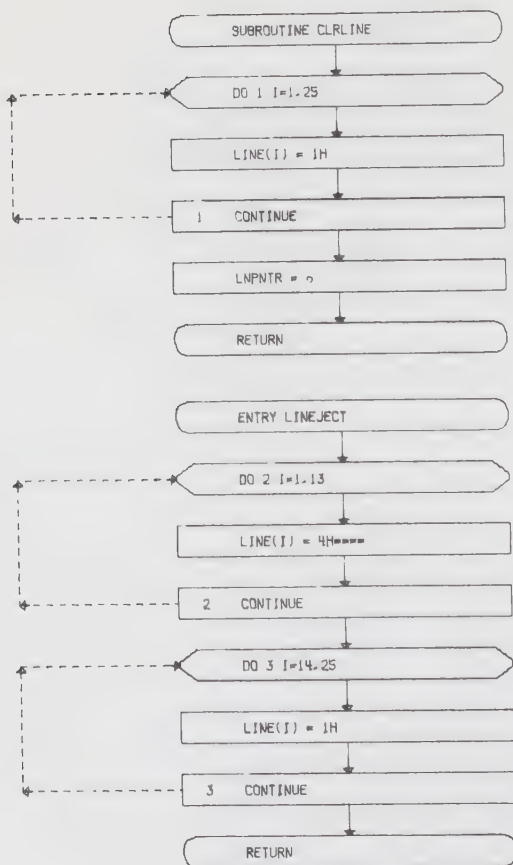
SUBROUTINE CLRLINE  
ENTRY LINEJECT

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /OUTLIST/ LNFNTR,LINE(25)

SUBROUTINE CLRLINE

1

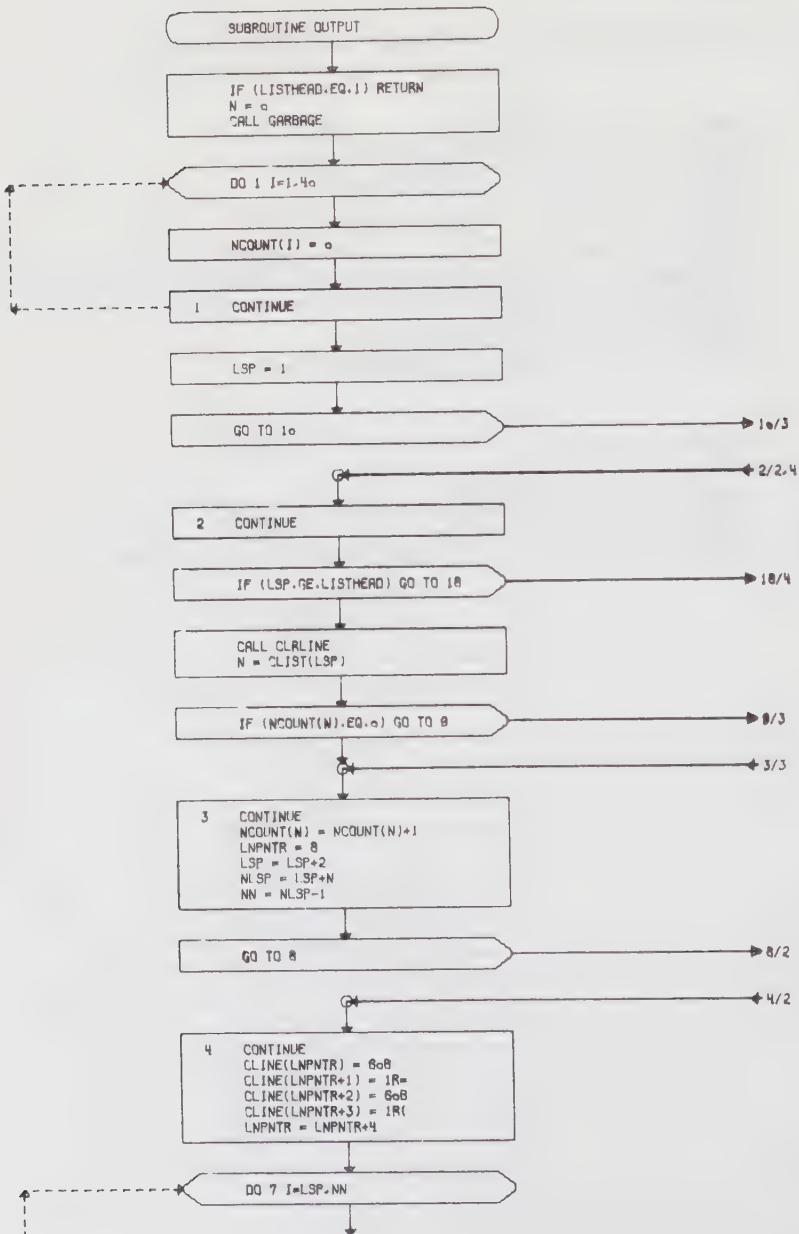


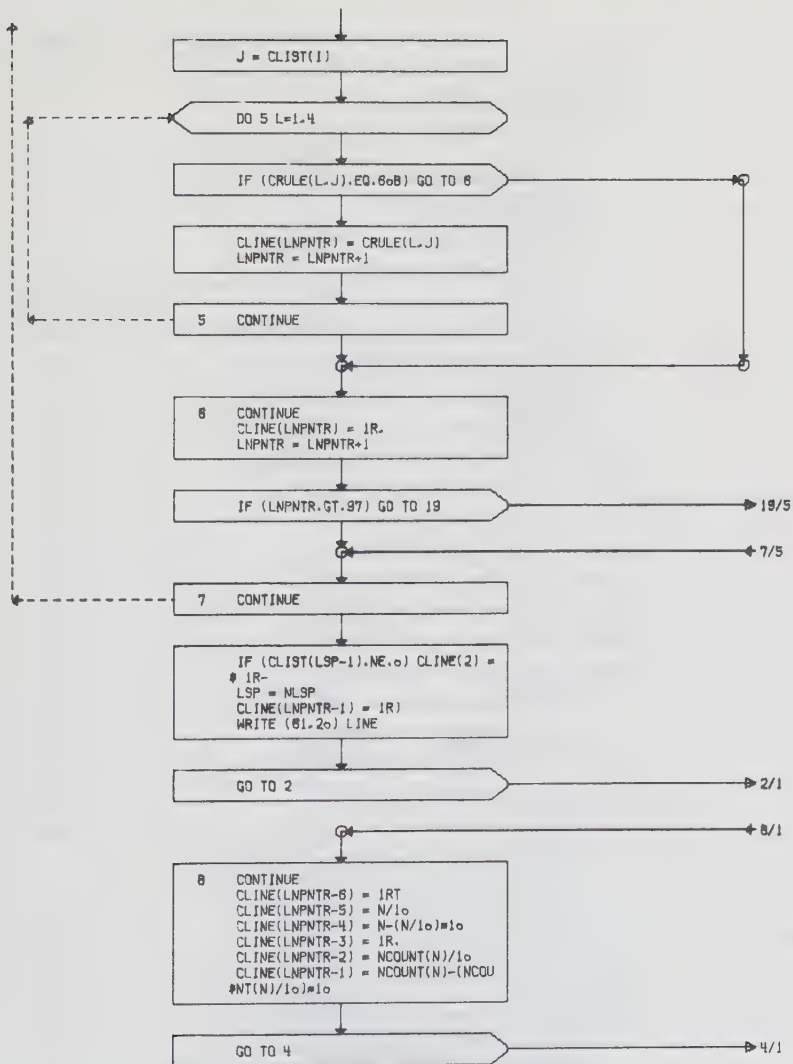
SUBROUTINE OUTPUT

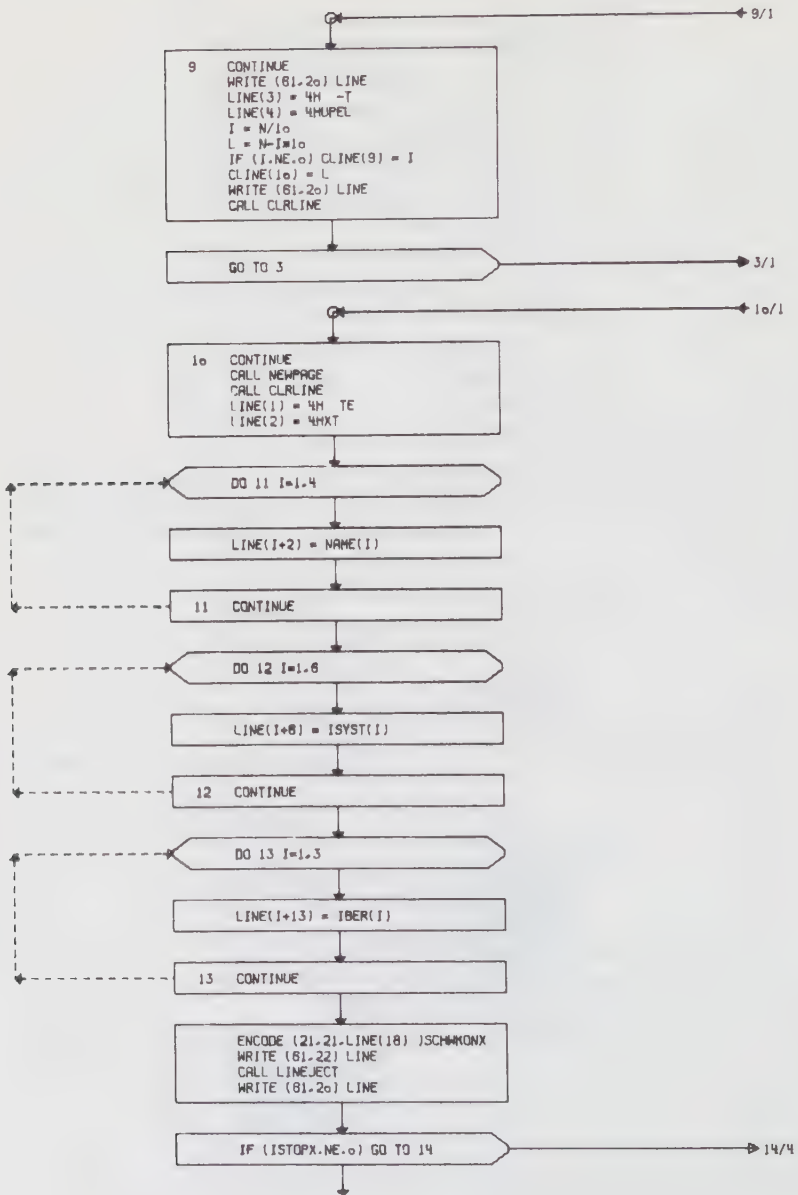
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

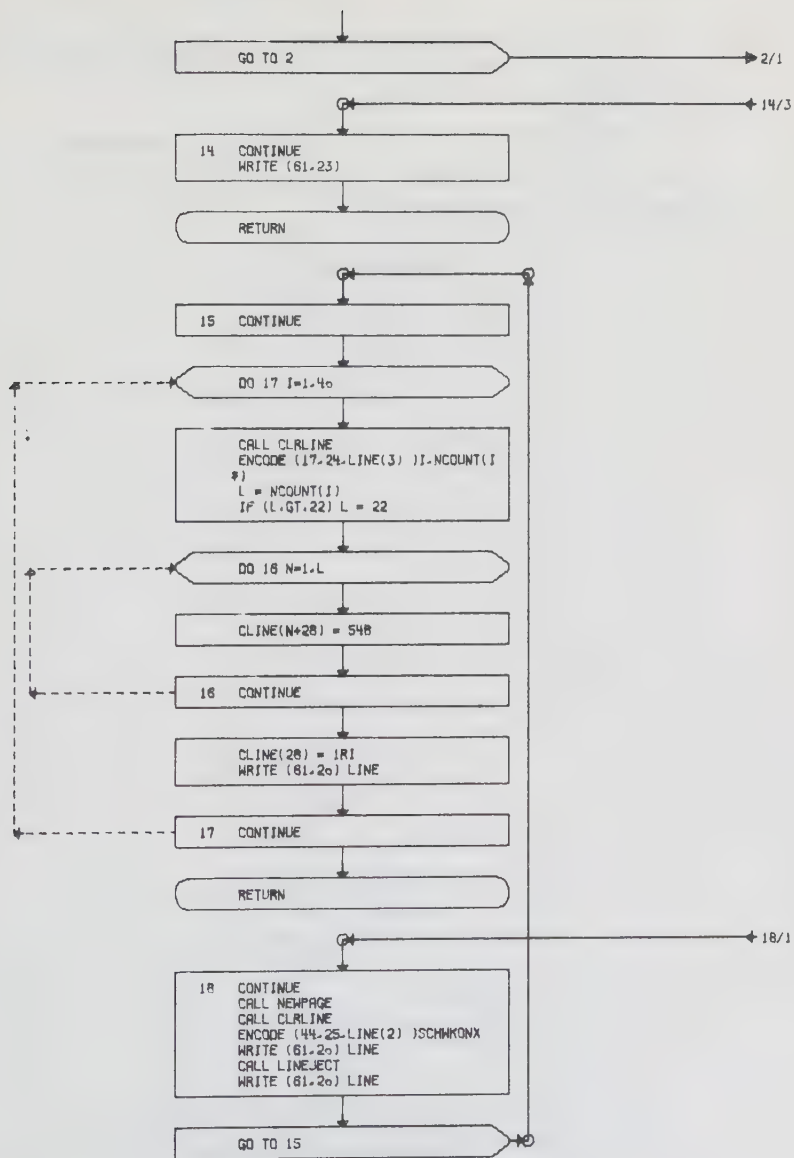
```
COMMON /STOPX/ ISTOPX
COMMON /OUTLIST/ LNPTR,LINE(25)
COMMON /LST/ LISTHEAD,LSTLENG
CHARACTER CRULE(4,40),CLINE(100)
EQUIVALENCE (CRULE,LRULE), (LINE,CLINE)
INTEGER CLIST
COMMON /X/ NAME(10),LRULE(40),IRPMAX,IVEND,IRPOINT(64),EUKLABST(40
$,40),RMATRIX1(40,40),RMATRIX2(40,40),RSUMPROZ(40),NSUM(64),JSUM
COMMON /KONNEX/ IKONNEX(10,40),SCHWKONX
DIMENSION NCOUNT(40)
COMMON /TITLE/ IBER(3),ISYST(6)
COMMON /TUPEL/ N,IND(40),NGARBFLG
```

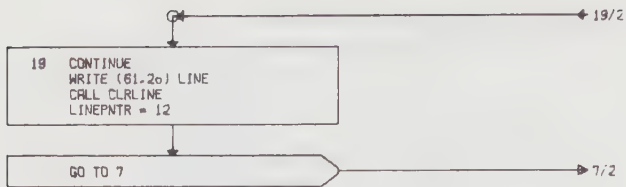
```
20 FORMAT (20X,25A4)
21 FORMAT(* SCHWELLWERT = *,F5.2)
22 FORMAT (20X,12A4/20X,13A4)
23 FORMAT(20X,10(1H*),* SCHWELLWERT ZU GROSS *,10(1H*))
24 FORMAT (I3,*-TUPEL*,1X,I5)
25 FORMAT(*VERTEILUNG DER N-TUPEL BEI SCHWELLWERT*,F6.2)
```











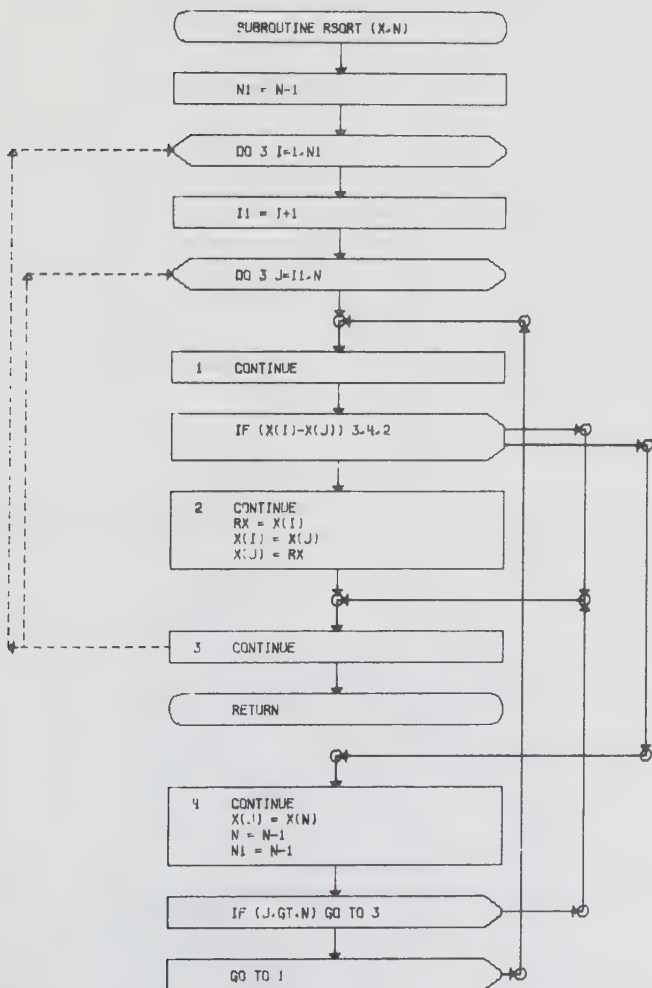
SUBROUTINE RSORT (X,N)

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

DIMENSION X(N)

SUBROUTINE RSORT

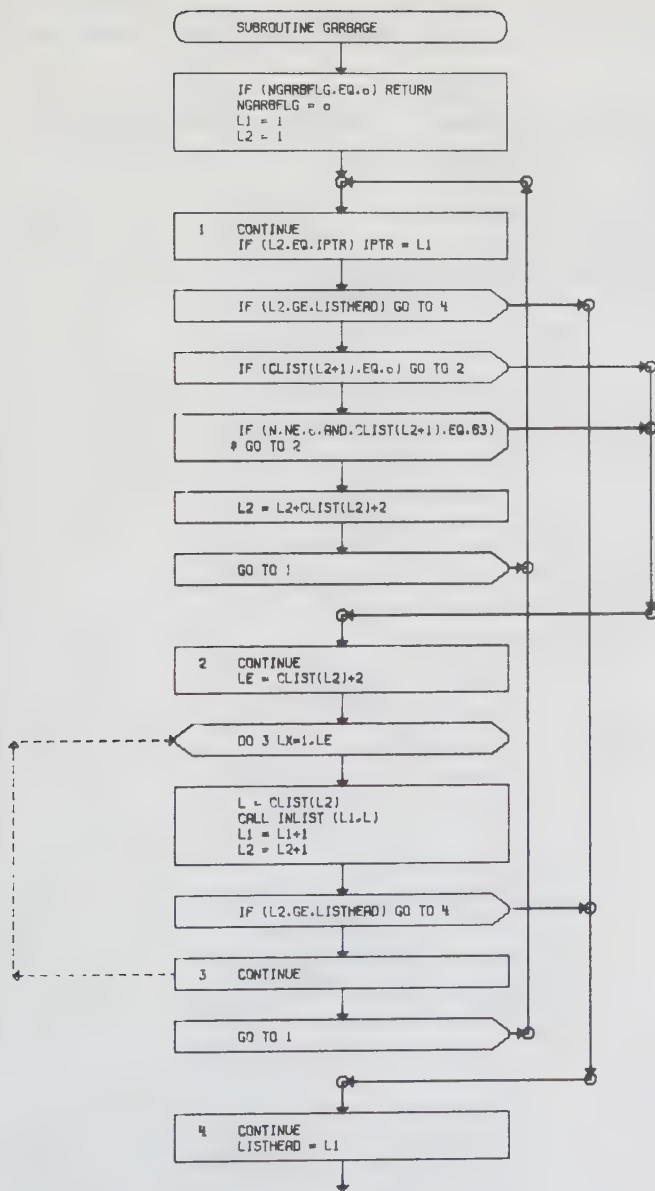
1

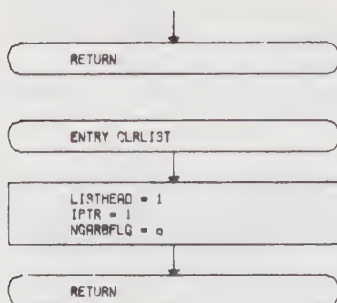


SUBROUTINE GARBAGE  
ENTRY CLRLIST

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /TUPEL/ N,NTUPLE(40),NGARBFLG  
COMMON /LST/ LISTHEAD,LSTLENG  
COMMON /PTR/ IPTR  
INTEGER CLIST





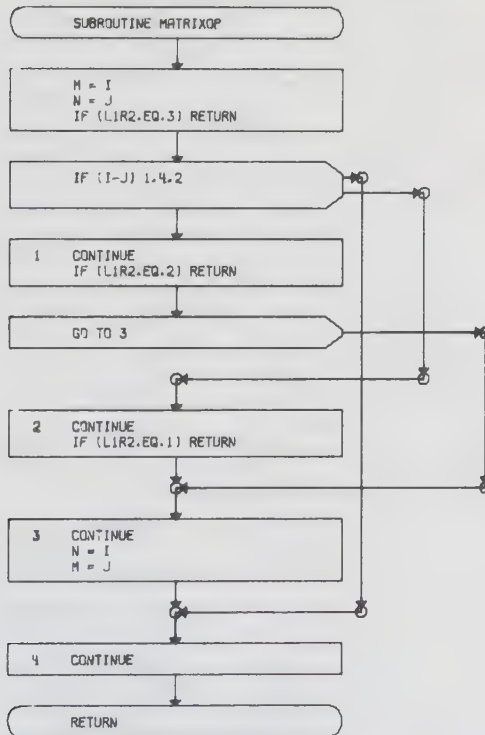
SUBROUTINE MATRIXOP

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /PAR/ I,J,L1R2,M,N

SUBROUTINE MATRIXOP

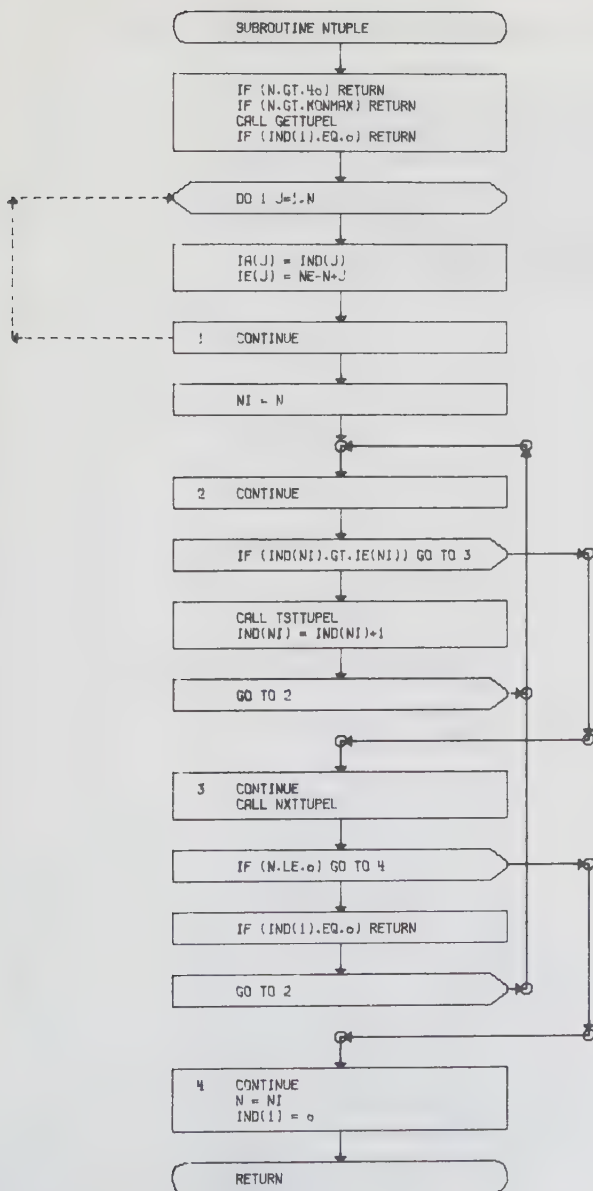
1



SUBROUTINE NTUPLE

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

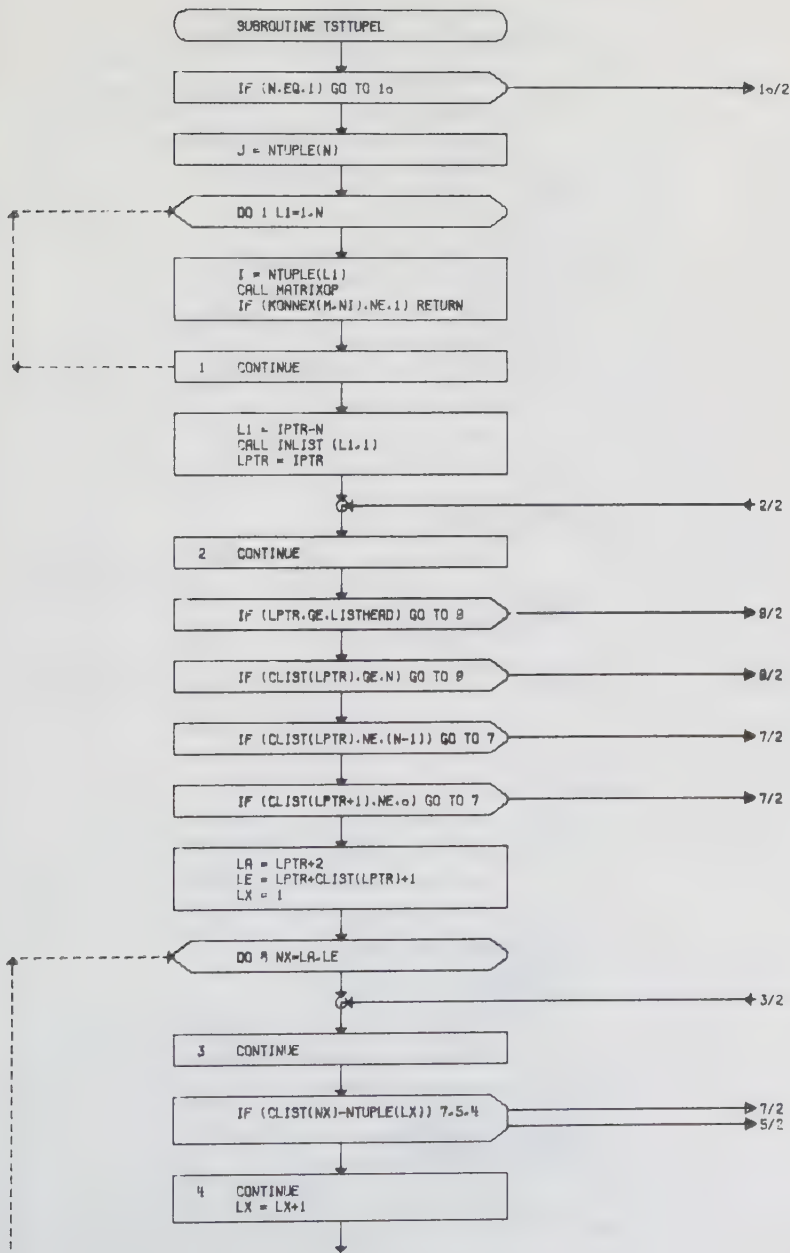
COMMON /TUPEL/ N,IND(40),NGARBFLG  
DIMENSION IA(40), IE(40)  
COMMON /LIMITS/ NA,NE,KONMAX,XMAX,YMAX,RMAX

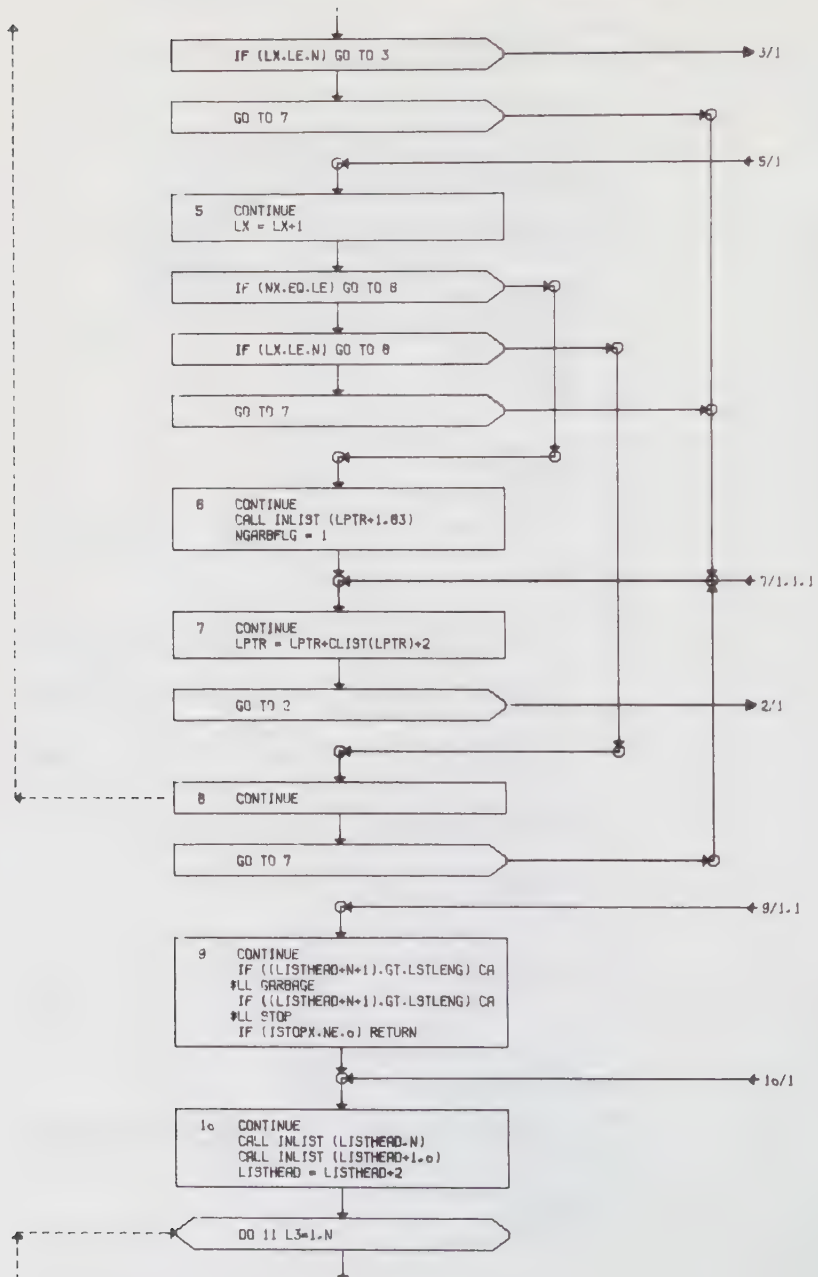


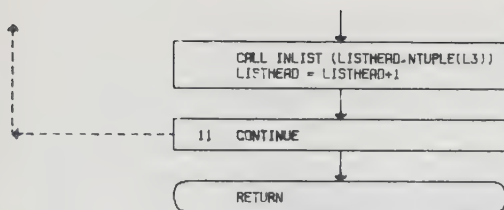
SUBROUTINE TSTTUPEL

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON /STOPX/ ISTOPX
COMMON /PAR/ I,J,L1R2,M,NI
COMMON /TUPEL/ N,NTUFLE(40),NGARBFLG
INTEGER CLIST
COMMON /LST/ LISTHEAD,LSTLENG
COMMON /KONNEX/ IKONNEX(10,40),SCHWKONX
CHARACTER KONNEX(40,40)
EQUIVALENCE (IKONNEX,KONNEX)
COMMON /PTR/ IPTR
```



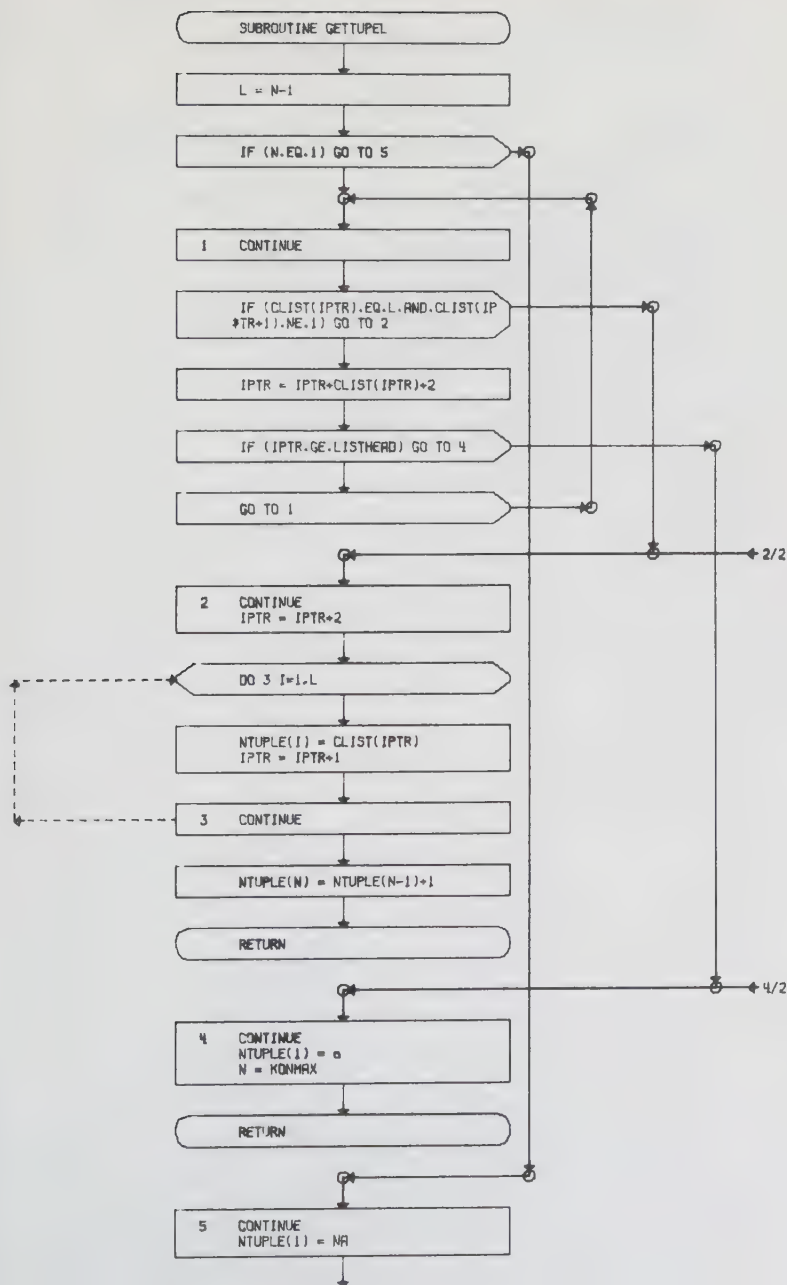


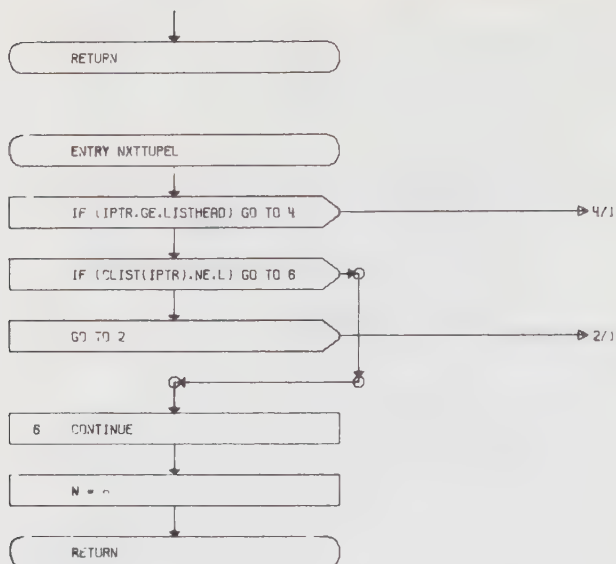


SUBROUTINE GETTUPEL  
ENTRY NXTTUPEL

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /LIMITS/ NA,NE,KONMAX,XMAX,YMAX,RMAX  
COMMON /TUPEL/ N,NTUPLE(40),NGARBFLG  
COMMON /LST/ LISTHEAD,LSTLENG  
INTEGER CLIST  
COMMON /PTR/ IPTR





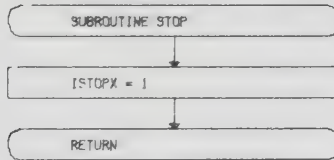
SUBROUTINE STOP

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /STOPX/ ISTOPX

SUBROUTINE STOP

1

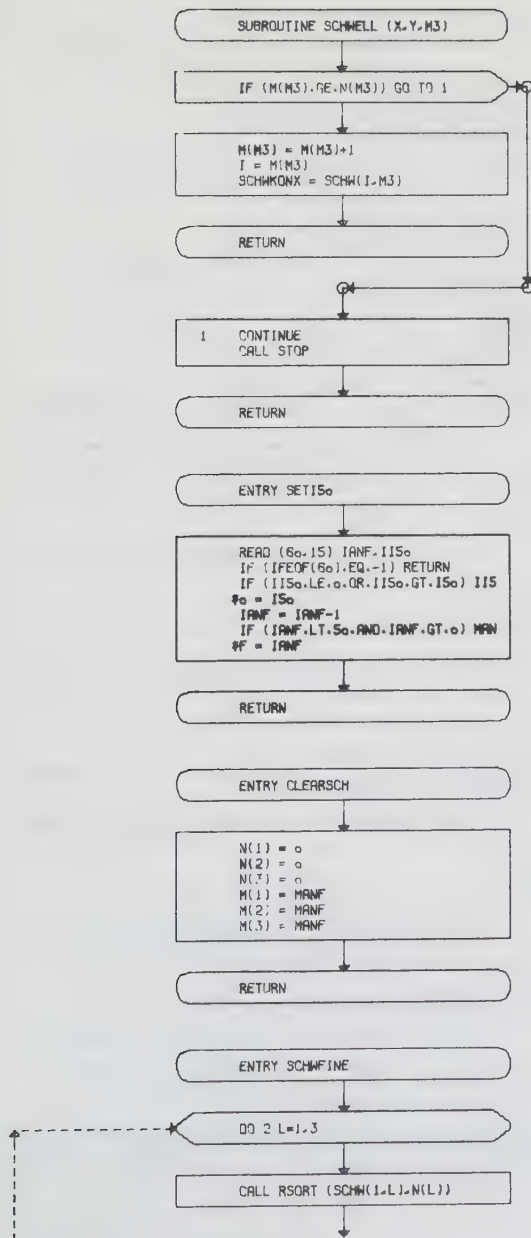


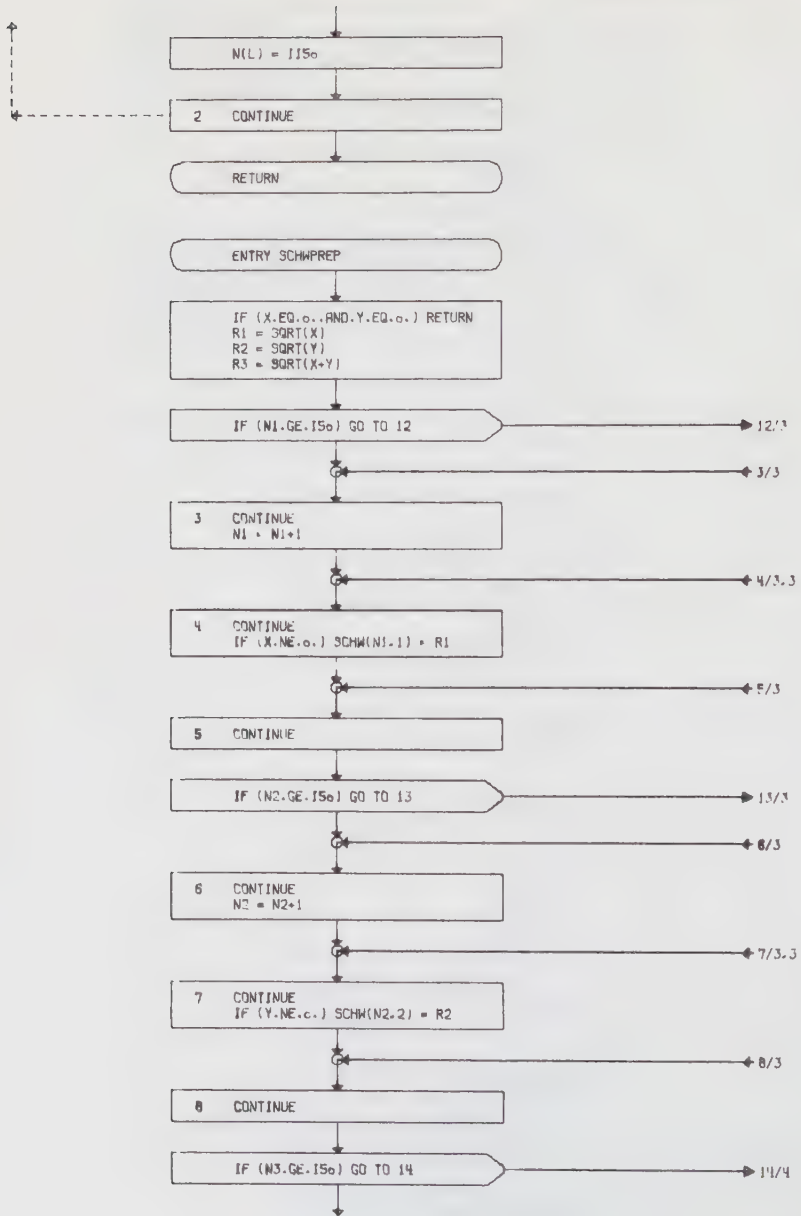
```
SUBROUTINE SCHWELL (X,Y,M3)
ENTRY SETI50
ENTRY CLEARSC
ENTRY SCHWFINE
ENTRY SCHWPREP
```

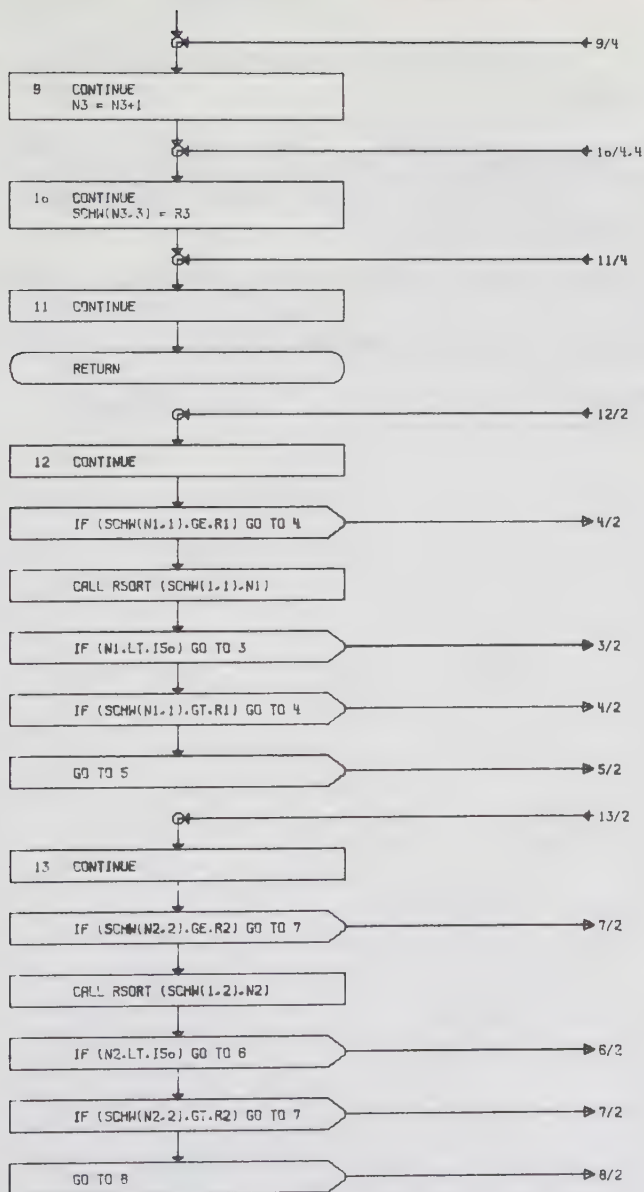
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON /KONNEX/ IKONNEX(10,40),SCHWKONX
DIMENSION SCHW(50,3), N(3), M(3)
EQUIVALENCE (N(1),N1), (N(2),N2), (N(3),N3)
DATA I50/50/
DATA MANF/0/
```

15   FORMAT (2I10)

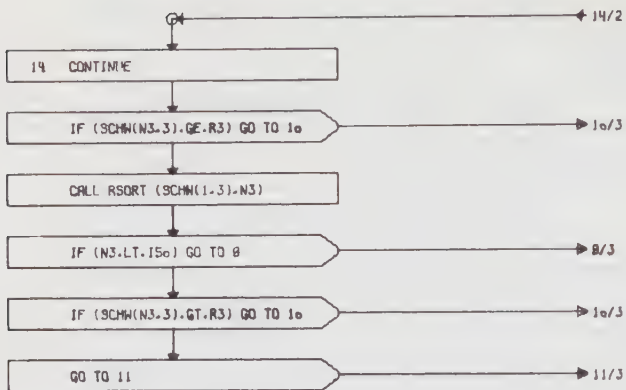






SUBROUTINE SCHWELL

4



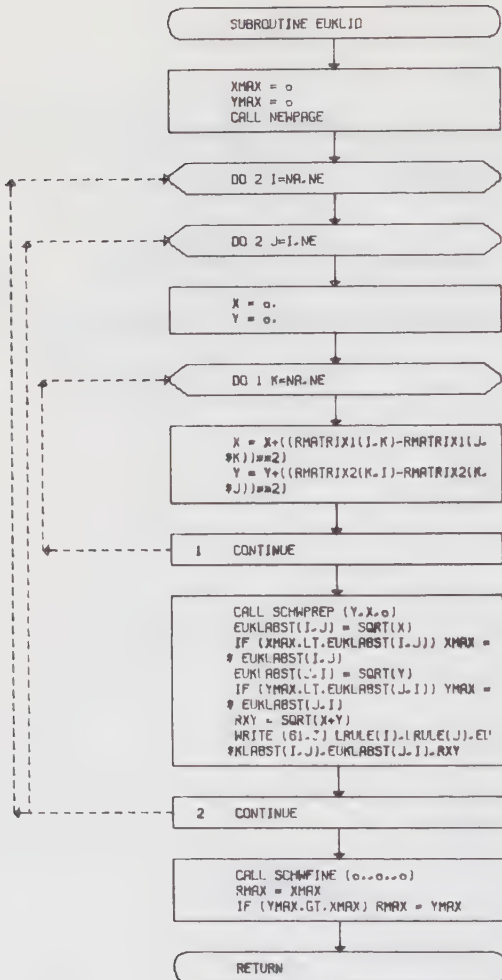
SUBROUTINE EUKLID

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /X/ NAME(10),LRULE(40),IRPMAX,IVEND,IRPOINT(64),EUKLABST(40  
\$,40),RMATRIX1(40,40),RMATRIX2(40,40),RSMUPROZ(40),NSUM(64),JSUM  
COMMON /LIMITS/ NA,NE,KONMAX,XMAX,YMAX,RMAX

SR≤I 7== R≤I#Z=7G=≤ \$P3[PX6P3[P\$ I=R≤I=G H== =I[77≤I\$PX2P7[=H99 4

\$EICH\$\*,F10.3,\* ,DES VORBEREICH\$\*,F10.3,\* , DER UMGEBUNG\*,F10.3  
\$)

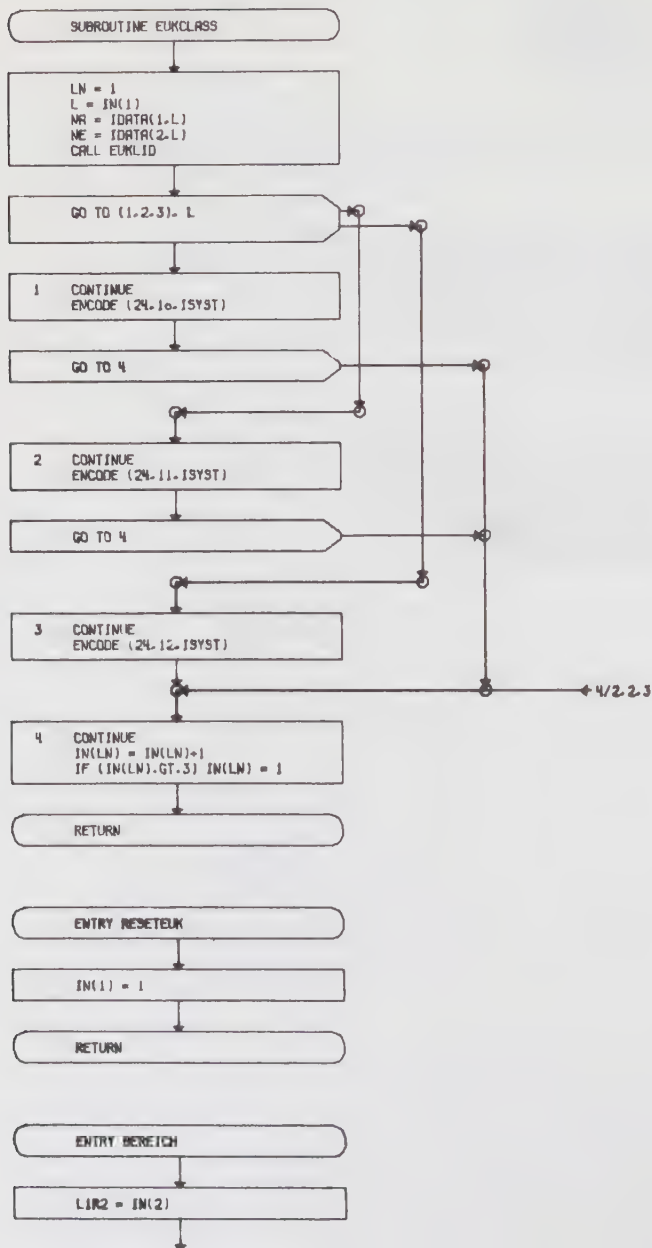


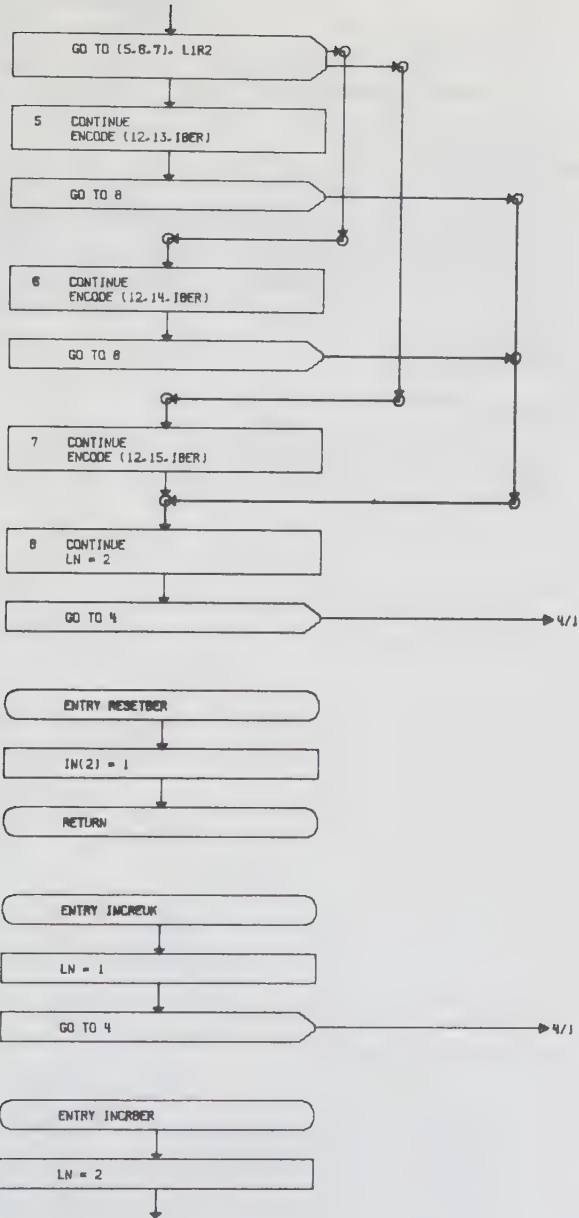
```
SUBROUTINE EUKCLASS
ENTRY RESETEUK
ENTRY BEREICH
ENTRY RESETBER
ENTRY INCREUK
ENTRY INCRBER
ENTRY CLASS
```

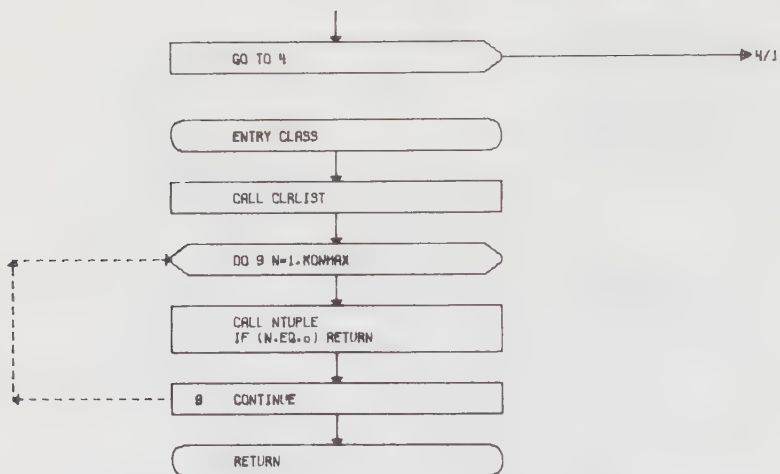
C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

```
COMMON /TUPEL/ N,NTUPL(40),NGARBFLG
COMMON /LIMITS/ NA,NE,KONMAX,XMAX,YMAX,RMAX
COMMON /DATA/ IDATA(2,3)
COMMON /PAR/ I,J,L1R2,M,NI
COMMON /TITLE/ IBER(3),ISYST(6)
DIMENSION IN(2)
DATA IN/2*1/
```

```
10 FORMAT(*SYSTEM DER GRAPHEME 2 *)
11 FORMAT(*SYSTEM DER GRAPHEME 1 *)
12 FORMAT(*SYSTEM DER GRAPHEME *)
13 FORMAT(*VORBEREICH *)
14 FORMAT(*NACHBEREICH *)
15 FORMAT(*UMGEBUNG *)
```



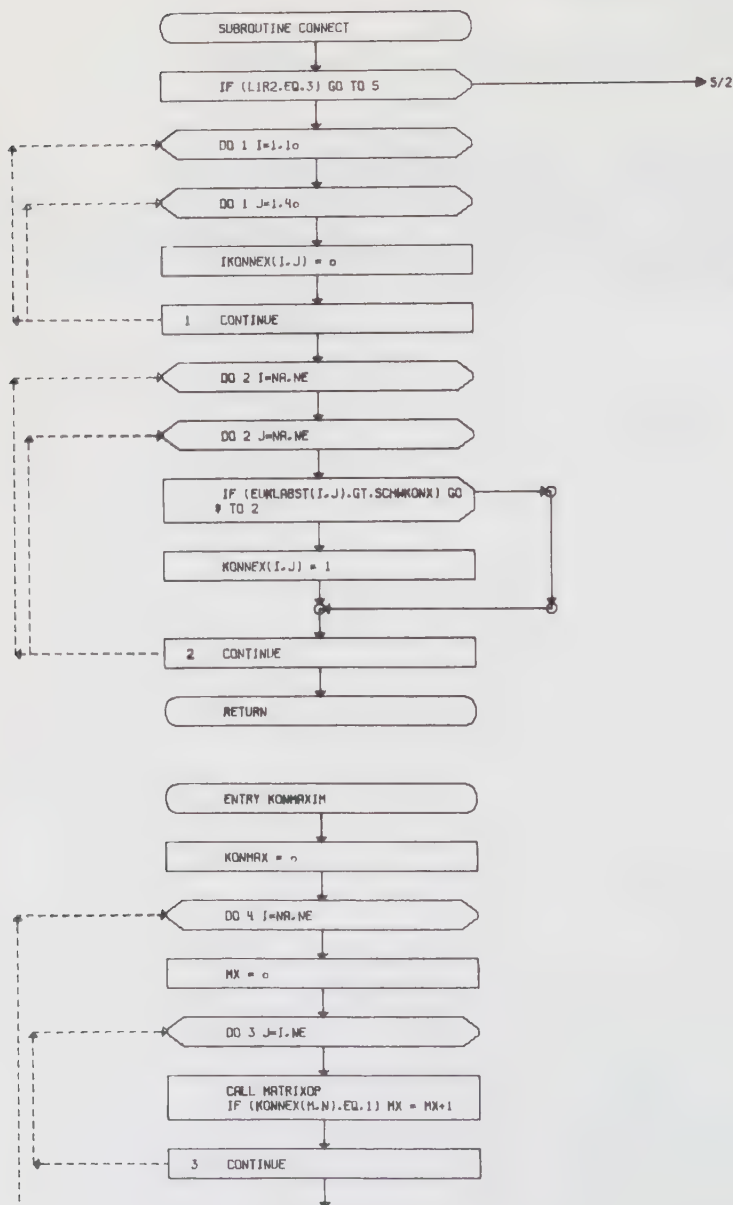


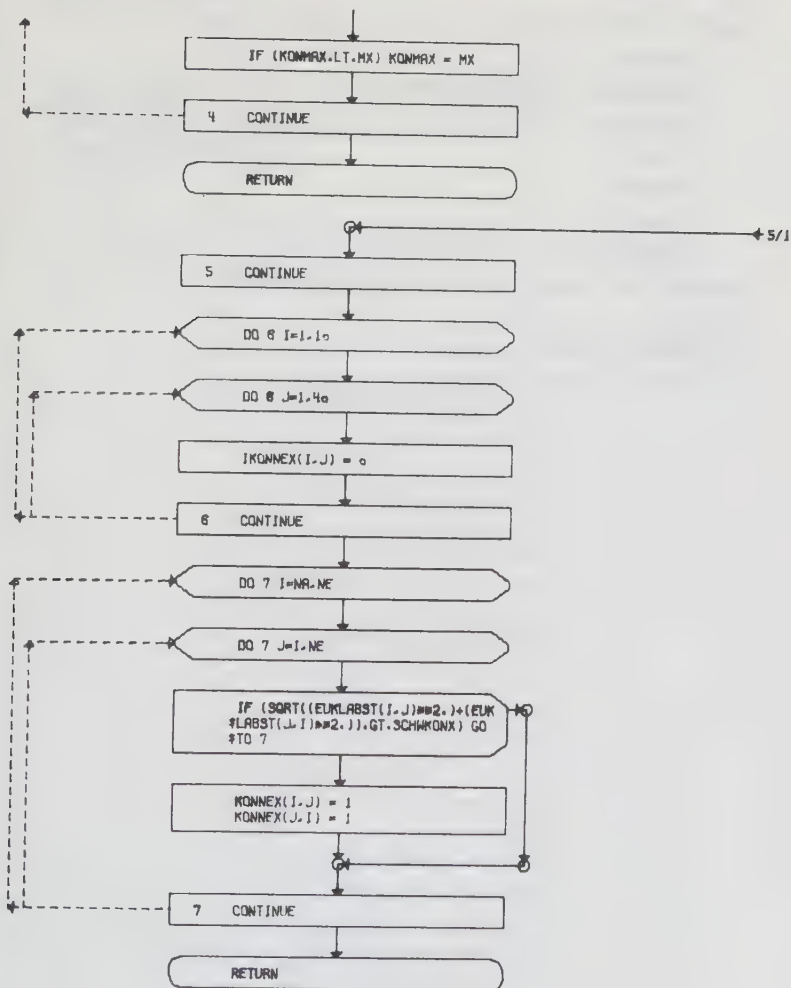


SUBROUTINE CONNECT  
ENTRY KONMAXIM

C \*\*\*\*\* SINGLE WORD INTEGER OPTION \*\*\*\*\*

COMMON /LIMITS/ NA,NE,KONMAX,XMAX,YMAX,RMAX  
COMMON /KONNEX/ IKONNEX(10,40),SCHWKONX  
CHARACTER KONNEX(40,40)  
EQUIVALENCE (IKONNEX,KONNEX)  
COMMON /X/ NAME(10),LRULE(40),IRPMAX,IVEND,IRPOINT(64),EUKLABST(40  
\$,40),RMATRIX1(40,40),RMATRIX2(40,40),RSUMPROZ(40),NSUM(64),JSUM  
COMMON /PAR/ I,J,L1R2,M,N





### 3.4.13 Programmbeschreibung WSEG

Das Programm WSEG besteht aus folgenden Teilprogrammen:

SILBTREN (Hauptprogramm)

RRULEGEN

SILBE

RTRANS

TRANSL

AUSGABE

Zusätzlich werden folgende Unterprogramme verwendet:

ISCAN (siehe 3.4.01)

MOVEC (siehe 3.4.01).

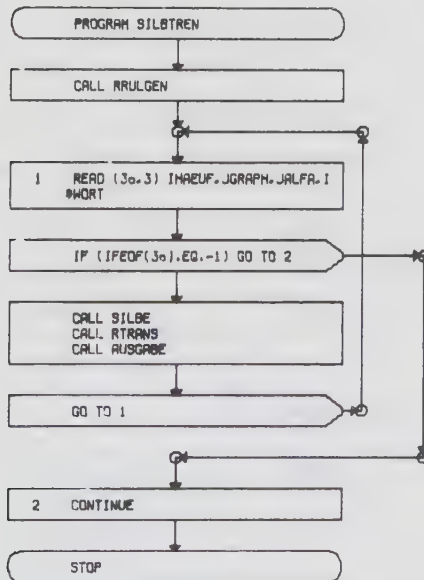
PROGRAM SILBTREN

```
COMMON /WORT/ JALFA(5),IWORT(5),ISILB(4,8),IKETT(5,8),IS1,IKONSTIT  
$(8),IHAEUF  
COMMON /RRULE/ IRULE(64)  
DIMENSION JGRAPH(5)
```

3     FORMAT (1X,I7,2X,5A8,3X,5A8,3X,5A8)

PROGRAM SILBTREN

1



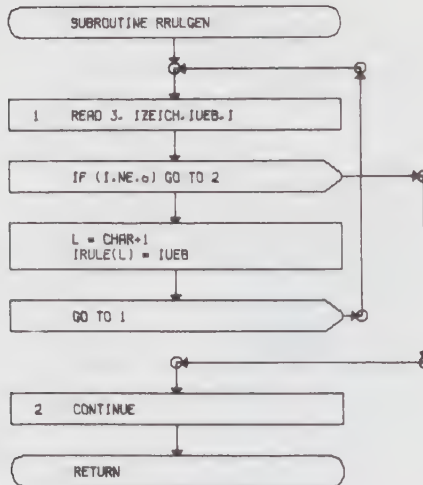
SUBROUTINE RRULGEN

COMMON /RRULE/ IRULE(64)  
CHARACTER CHAR  
EQUIVALENCE (CHAR, IZEICH)

3     FORMAT (A1,A8,I2)

SUBROUTINE RRULGEN

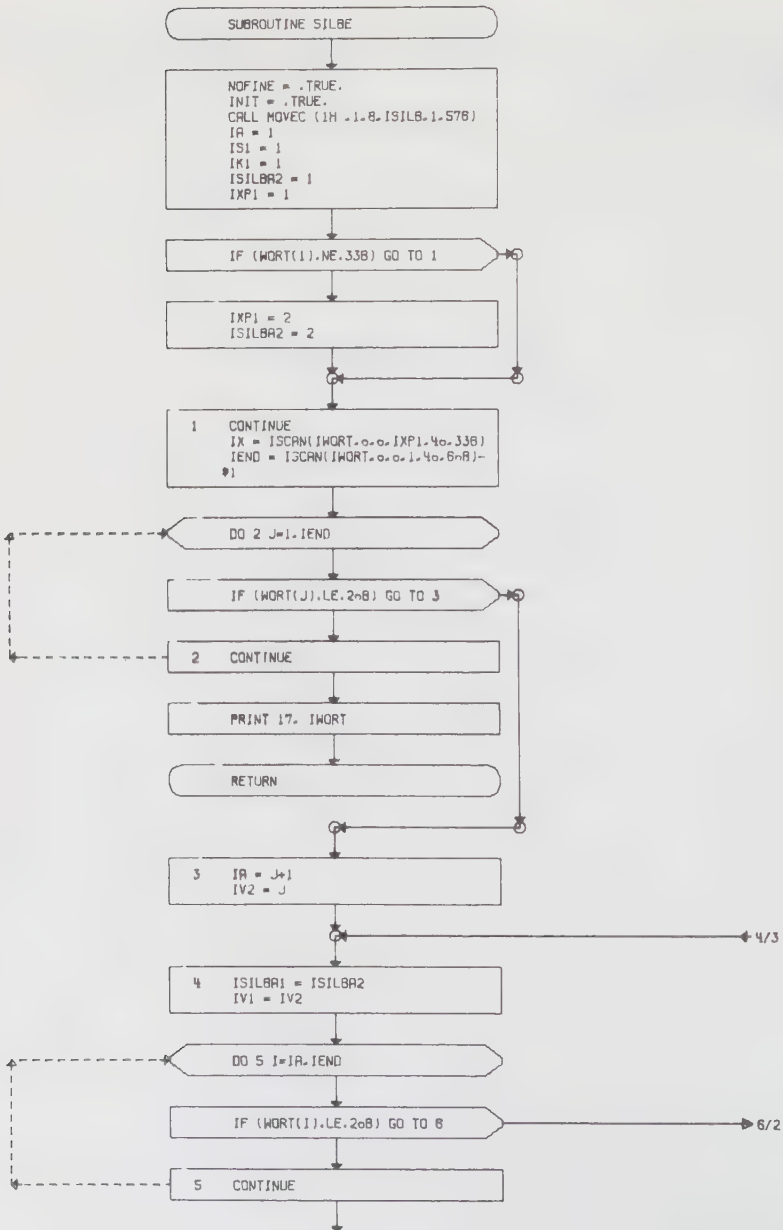
1

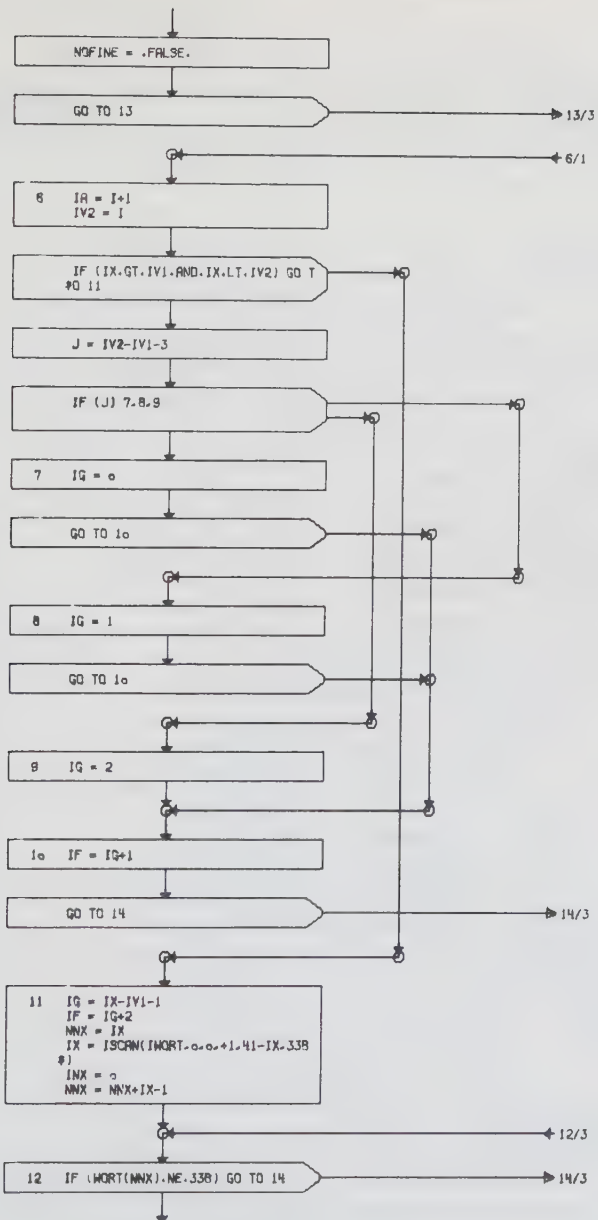


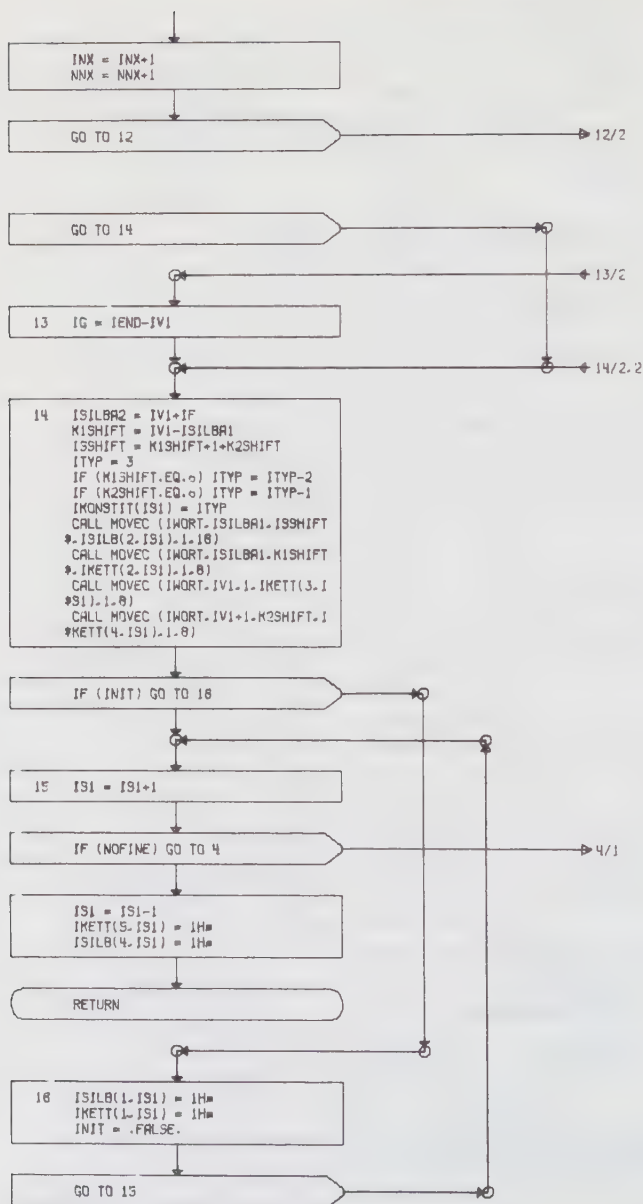
SUBROUTINE SILBE

```
COMMON /WORT/ JALFA(5),IWORT(5),ISILB(4,8),IKETT(5,8),IS1,IKONSTIT  
$(8),IHAEUF  
LOGICAL INIT,NOFINE  
CHARACTER WORT(40)  
EQUIVALENCE (IG,K2SHIFT), (IWORT(1),WORT(1))
```

17   FORMAT(\* KEIN GRAPHEN VOM TYP 1 IN \*,5A8)





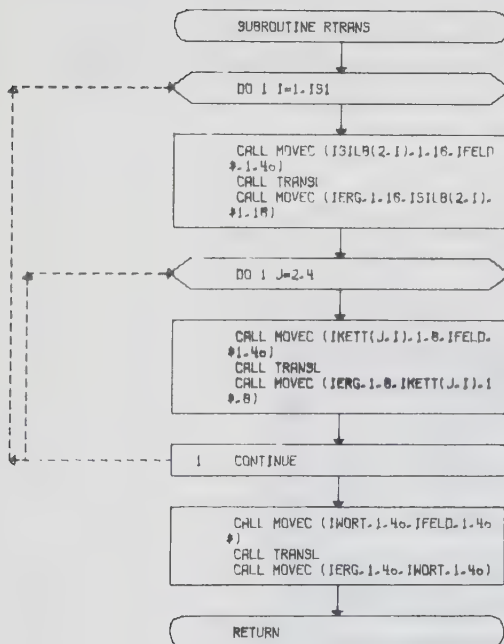


SUBROUTINE RTRANS

COMMON /WORT/ JALFA(5),IWORT(5),ISILB(4,8),IKETT(5,8),IS1,IKONSTIT  
\$(8),IHAEUF  
COMMON /RTRFLO/ IFELD(5),IERG(5)

SUBROUTINE RTRANS

1

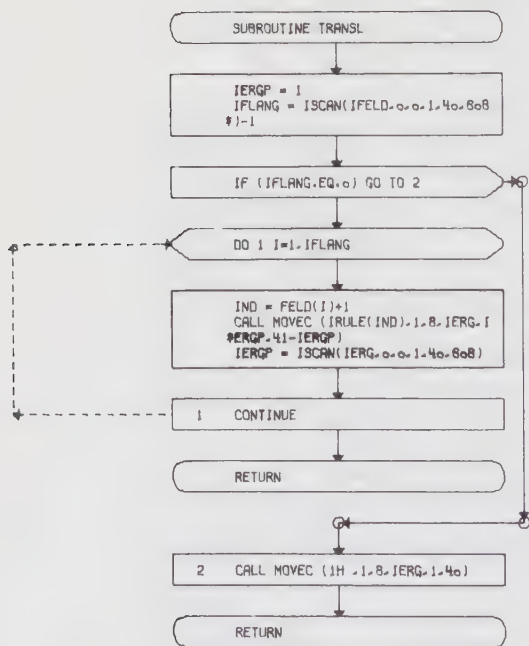


# SUBROUTINE TRANSL

```
COMMON /RTRFLD/ IFELD(5),IERG(5)
COMMON /RRULE/ IRULE(64)
CHARACTER FELD(40)
EQUIVALENCE (IFELD(1),FELD(1))
```

SUBROUTINE TRANSL

1



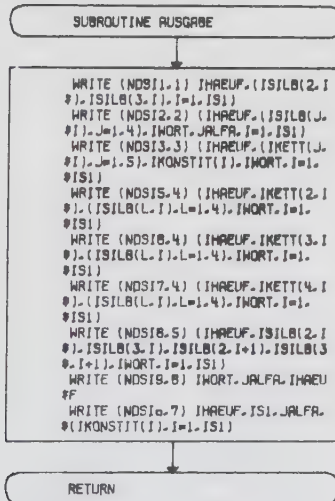
# SUBROUTINE AUSGABE

```
COMMON /WORT/ JALFA(5),IWORT(5),ISILB(4,8),IKETT(5,8),IS1,IKONSTIT
$(8),IHAEUF
DATA NDSI1/40/,NDSI2/41/,NDSI3/42/,NDSI5/44/,NDSI6/45/,NDSI7/46/,
$NDSI8/47/,NDSI9/48/,NDSI0/49/
```

```
1  FORMAT (1X,I5,2X,8(A8,A1))
2  FORMAT (1X,I7,3X,A2,2A8,A2,4X,5A8,4X,5A8)
3  FORMAT (1X,I5,2X,A2,3A8,A2,I2,2X,5A8)
4  FORMAT (4X,I7,4X,A8,4X,A2,2A8,A2,4X,5A8)
5  FORMAT (4X,I7,10X,2A8,4X,2A8,4X,5A8)
6  FORMAT (4X,5A8,4X,5A8,5X,I7)
7  FORMAT (3X,I7,2X,I2,3X,5A8,7X,8I2)
```

SUBROUTINE AUSGABE

1



#### 3.4.14 Programmbeschreibung WKST

Das Programm WKST besteht aus einem FORTRAN-Programm.

Für die Version ohne Belege ist dies:

WKSSTAT1 (Hauptprogramm).

Für die Version mit Belegzitaten ist dies:

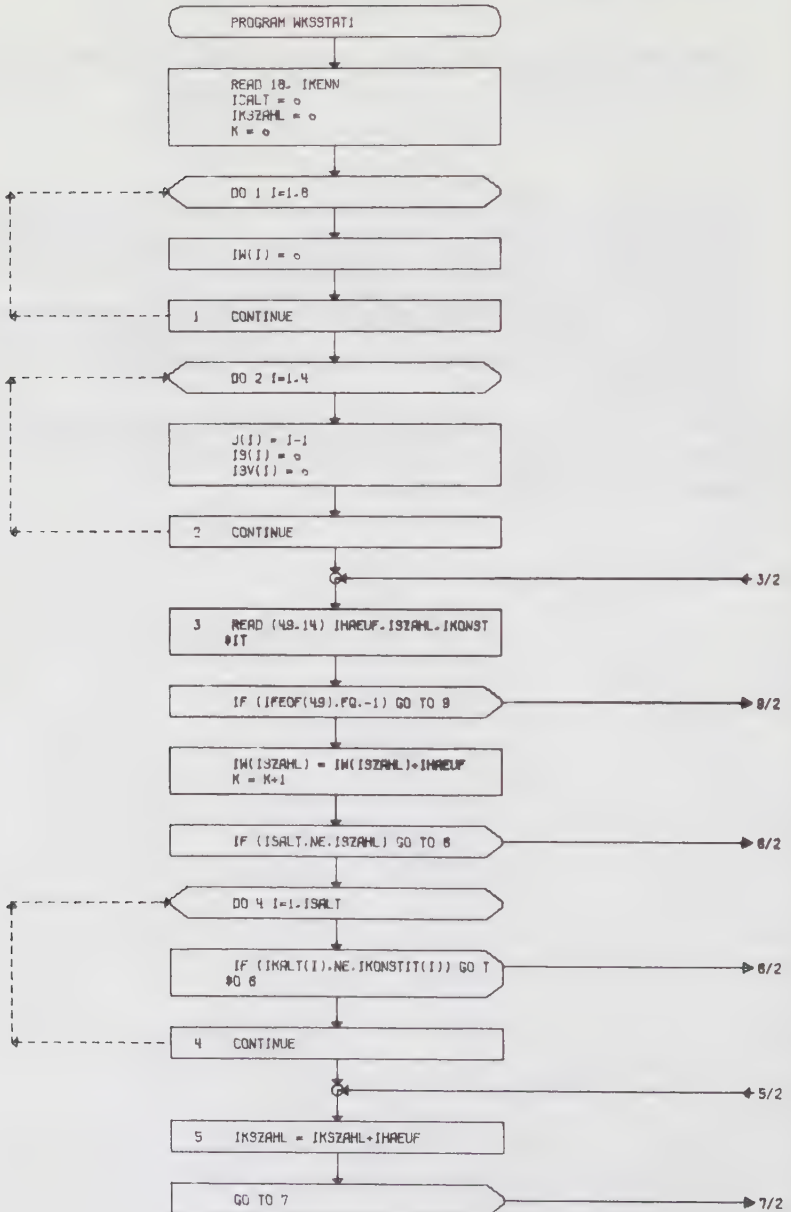
WKSSTAT2 (Hauptprogramm).

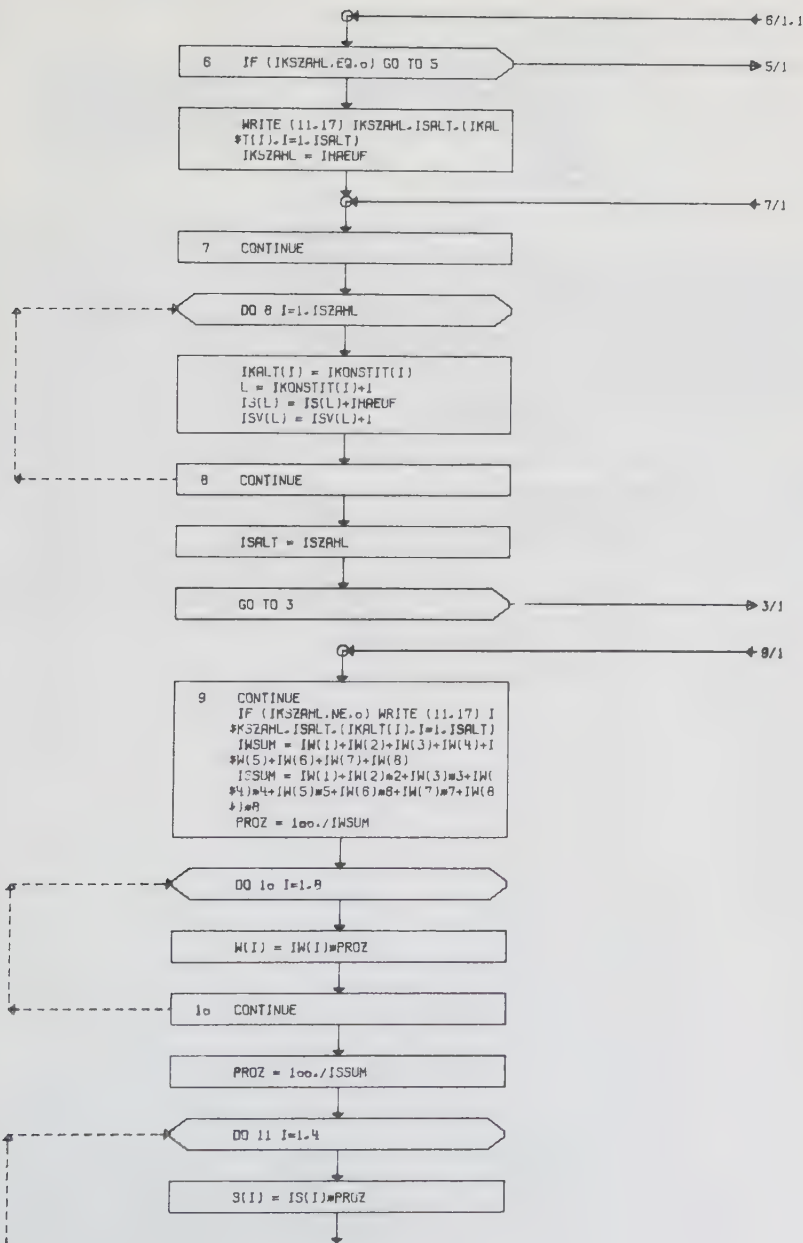
PROGRAM WKSSTAT1

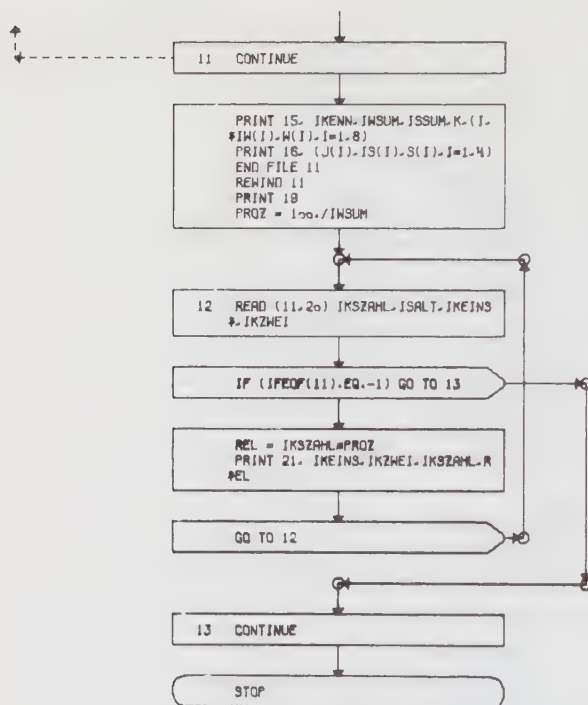
DIMENSION IKONSTIT(8), IW(8), IS(4), W(8), S(4), ISV(4), J(4)  
 DIMENSION IKENN(5)  
 DIMENSION IKALT(8)

```

14  FORMAT (3X,I7,2X,I2,50X,8I2)
15  FORMAT (1H1,29X,13HTEXTKENNWERTE/30X,13(1H*)/1H-,12X,4HTEXT,2X,A8,
    $5X,4A8/1H0,12X,26HZAHL DER WOERTER      W = ,I7/1H ,12X,26HZAHL DE
    $R SEGMENTE      S = ,I7/1H ,12X,26HZAHL DER WORTFORMEN  F = ,I7/
    $1H-,22X,36HVERTEILUNG DER SEGMENTZAHL/WORT L(W)/1H ,22X,36(1H-)/
    $14X,7HRELATIV/1H /(1H ,12X,I4,10X,I7,14X,F7.4))
16  FORMAT (1H-,20X,39HHAUEFIGKEITSVERTEILUNG DER SEGMENTTYPEN/1H ,20X
    $,39(1H-)/1H0,10X,*SEGMENTTYP*,11X,21HBELEGHAUEFIGKEIT P(S)/1H ,27X
    $,7HABSOLUT,14X,7HRELATIV/1H /(1H ,12X,I4,10X,I7,14X,F7.4))
17  FORMAT (I5,I2,8I2)
18  FORMAT (5A8)
19  FORMAT (1H1,13X,42HVERTEILUNG DER WORTKONSTITUENTENSTRUKTUREN/1H ,
    $13X,42(1H-)/1H-,13X,8HSTRUKTUR,13X,4HZAHL,5X,7HRELATIV/1H )
20  FORMAT (I5,I2,2A8)
21  FORMAT (1H ,12X,2A8,5X,I5,5X,F7.4)
    
```



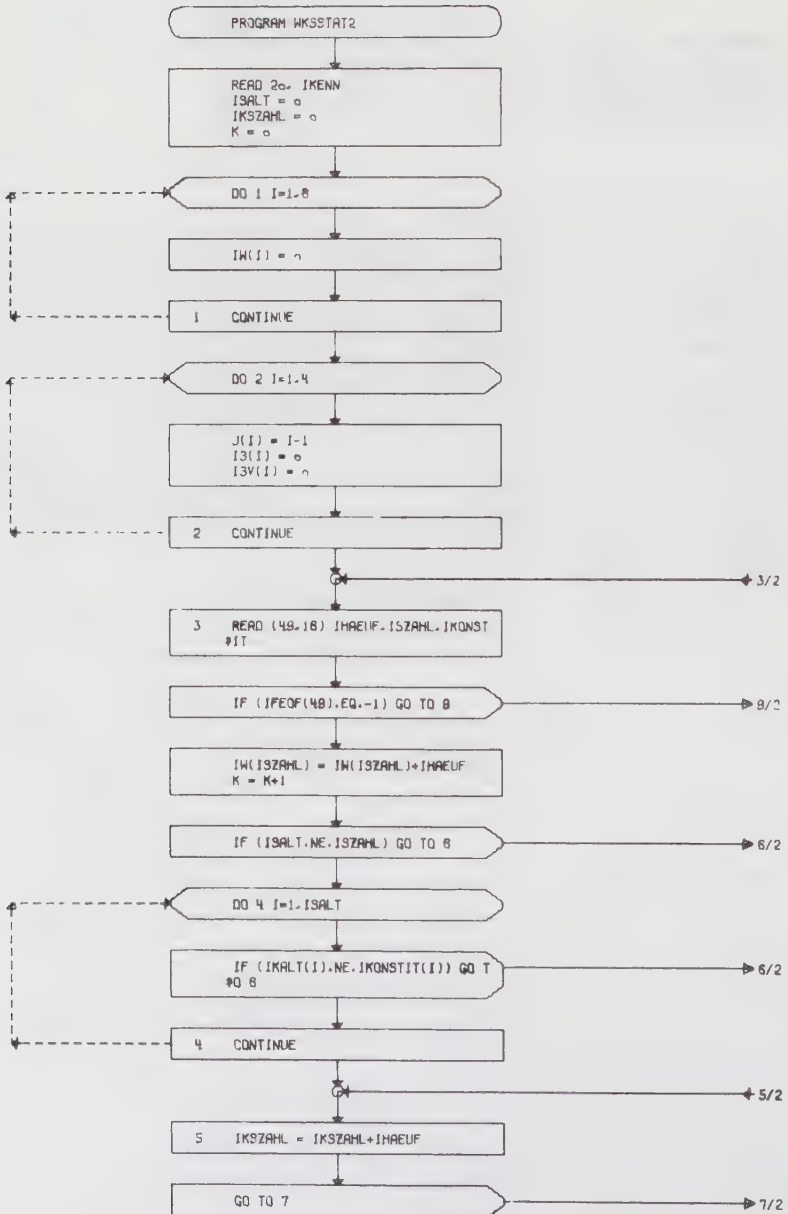


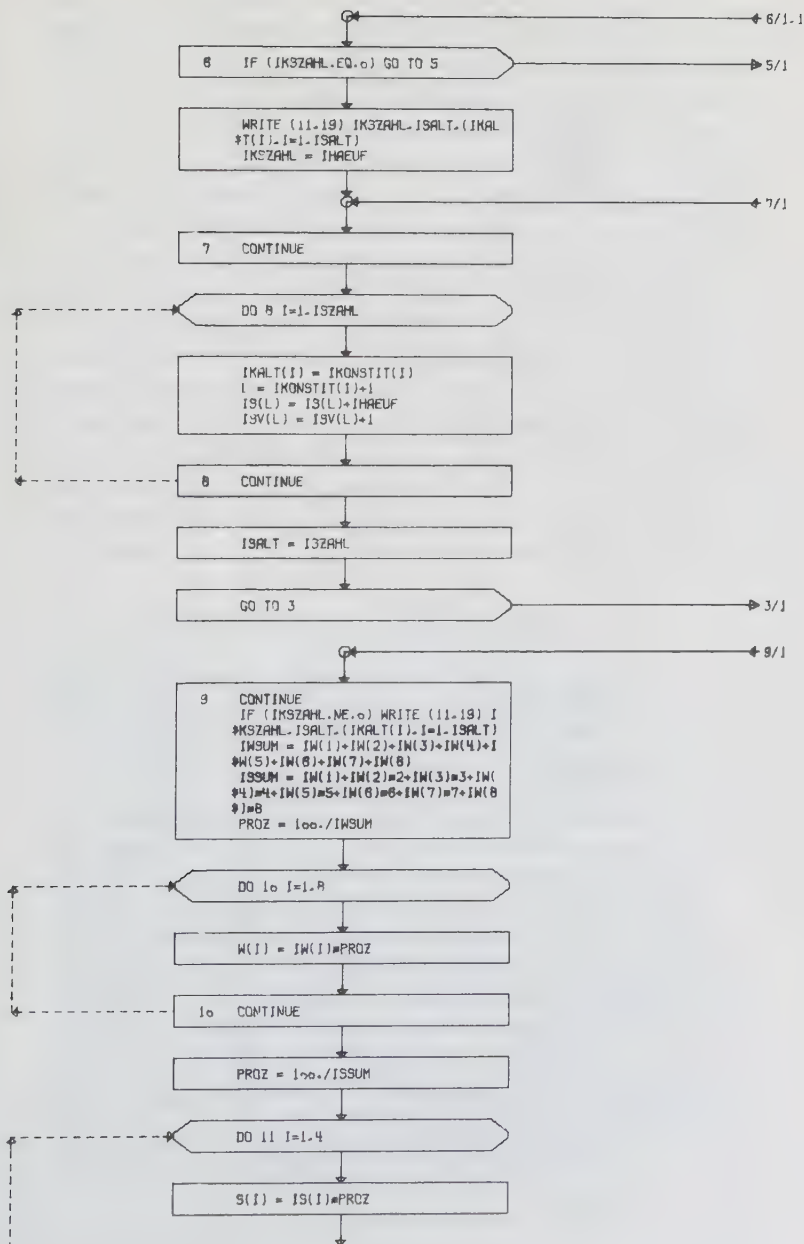


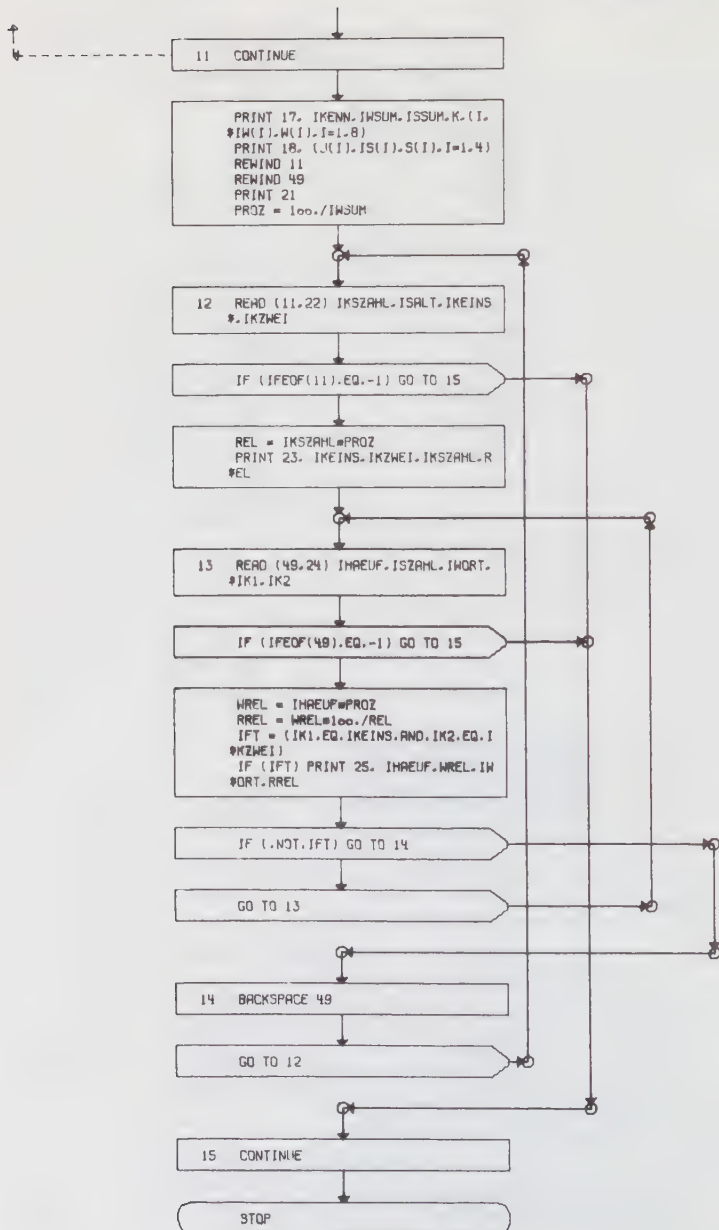
PROGRAM WKSSTAT2

```
DIMENSION IKONSTIT(3), IW(3), IS(4), W(3), S(4), ISV(4), J(4)
DIMENSION IKENN(5)
DIMENSION IKALT(8)
DIMENSION IWORT(5)
LOGICAL IFT
```

```
16  FORMAT (3X,I7,2X,I2,50X,8I2)
17  FORMAT (1H1,29X,13HTEXTKENNWERTE/30X,13(1H*)/1H-,12X,4HTEXT,2X,A8,
$5X,4A8/1H0,12X,26HZAHl DER WOERTER      W = ,I7/1H ,12X,26HZAHl DE
$R SEGMENTE      S = ,I7/1H ,12X,26HZAHl DER WORTFORMEN  F = ,I7/
$1H-,22X,36HVERTEILUNG DER SEGMENTZAHl/WORT L(W)/1H ,22X,36(1H-)/
$1H0,12X,4HL(W),12X,25HBELEGHAEUFIGKEIT P(L(W))/1H ,27X,7HABSOLUT,
$14X,7HRELATIV/1H /(1H ,12X,I4,10X,I7,14X,F7.4))
18  FORMAT (1H-,20X,39HHAUEFIGKEITSVERTEILUNG DER SEGMENTTYPEN/1H ,20X
$,39(1H-)/1H0,10X,*SEGMENTTYP*,11X,21HBELEGHAEUFIGKEIT P(S)/1H ,27X
$,7HABSOLUT,14X,7HRELATIV/1H /(1H ,12X,I4,10X,I7,14X,F7.4))
19  FORMAT (I5,I2,8I2)
20  FORMAT (5A8)
21  FORMAT (1H1,13X,42HVERTEILUNG DER WORTKONSTITUENTENSTRUKTUREN,
$18H MIT BELEGNACHWEIS/1H ,13X,42(1H-),18(1H-)/1H-,13X,8HSTRUKTUR,
$13X,4HZAHl,5X,7HRELATIV,10X,6HBELEGE,34X,11HREL(STRUKT)/1H )
22  FORMAT (I5,I2,2A8)
23  FORMAT (1H ,12X,2A8,5X,I5,5X,F7.4)
24  FORMAT (3X,I7,2X,I2,3X,5A8,7X,2A8)
25  FORMAT (1H ,36X,I5,5X,F7.4,7X,5A8,F8.4)
```







#### 4. Geplante und teilweise realisierte Erweiterungen

Das hier vorgestellte Programmsystem versteht sich als integrierter Bestandteil einer Programmbibliothek zur LDV. Diese Bibliothek wird gerade weiterentwickelt in Richtung auf syntaktische Analyse. In diesem Bereich gingen wir bisher den Weg der rechnergestützten Syntaxanalyse. Dazu existieren bereits Programme<sup>1</sup>, die auch schon auf umfangreiche Texte der Gegenwartssprache<sup>2</sup> angewandt wurden. Ein Konzept zur Erfassung der Strukturen im Dialog Mensch-Maschine ist gegenwärtig in Arbeit; dieses Projekt wird schon im Hinblick auf textlinguistische Ansätze entwickelt. Eine Einbettung semantischer Information ist ebenfalls vorgesehen. Dieses System wird auch zum Erstellen eines syntaktischen Wörterbuchs und zur Gewinnung eines syntaktischen Regelcorpus benutzt. Weiterhin ist ein syntaxgesteuerter Übersetzer (Compiler-Compiler)<sup>3</sup> zur leichten Modifikation von syntaktischen Analyse-Algorithmen in Arbeit.

---

<sup>1</sup>Vgl. etwa:

Böhm, S., Koller, G., Schönhut, J., Straßner, E., Komplexität von Sätzen als Kriterium für eine vergleichende Textanalyse, demnächst in: Linguistik und Didaktik.

Schönhut, J., Kurzbeschreibung eines Programmsystems zur Auswertung syntaktischer Beschreibungen auf der CD 3300/MASTER, interne Programmbeschreibung.

<sup>2</sup>Straßner, E., Schönhut, J., Sozio- und psycholinguistische Aspekte der Nachrichtensprache, Vortrag, gehalten am 25.9.1971 in Baden-Baden, abgedruckt in: Materialien zur Arbeitstagung "Sozio- und psycholinguistische Aspekte der Nachrichtensprache" veranstaltet von der Fachgruppe "Rundfunk und Sprache" in Zusammenarbeit mit der Hörfunkdirektion des Südwestfunks, Erlangen 1972. Eine überarbeitete Fassung dieses Vortrags demnächst in dem Sammelband "Sprache und Gesellschaft", (Fink-Verlag), München 1972.

<sup>3</sup>Schönhut, J., Die Verwendung von Compiler-Compilern zur Programmierung von Analysegrammatiken, Referat, gehalten am 12.11.1970 beim Kolloquium "Programmiersprachen für Linguistische Datenverarbeitung" beim Institut für deutsche Sprache in Mannheim.

## 5. Verzeichnis der Beispiel-Outputs<sup>1</sup>

- 1.1 Liste des Ausschnitts MS.Bodl.97o,317d, Zeile 32-48 (Listenausschnitt des Programms KONK).
- 1.2 Kopie vom Mikrofilm desselben Ausschnitts.
2. Relative Häufigkeit von A vor N als Funktion der Textlänge (BODL).
3. Kontextregister-Ausschnitt zu BODL (TYR bis UIL).
4. Komprimierte-Wortformen-Datei (gleicher Ausschnitt wie Beispiel 3).
5. Häufigkeitsregister der Wortformen (Ausschnitt 1-2o, BODL).
- 6.1 Ausschnitt aus komprimierten Wortformen vor "Praefix"-Abspaltung (BODL).
- 6.2 Gleicher Ausschnitt wie 6.1 nach "Praefix"-Abspaltung.
7. Ausschnitt aus Ketten (aus Ausschnitt aus MENT).
8. Ausschnitt aus den komprimierten Ketten (aus Ausschnitt wie Beispiel 7).
9. Ausschnitt aus Kettenübergangsliste (aus Ausschnitt wie Beispiel 7).
10. Ausschnitt aus dem vorläufigen Graphemsystem (BODL).
11. Ausschnitt aus dem Output des Programms GRUL (Regelsystem-Ausschnitt BODL).
- 12.1 Ausschnitt aus Graphemschreibungs-Datei (BODL).
- 12.2 Ausschnitt aus Anwendungs-Statistik der Regeln (BODL).
13. Standard-Rückübersetzungsregeln.
- 14.1 Häufigkeit der Grapheme (BODL).
- 14.2 Verteilung der Länge der Wortformen (Grapheme in BODL).
- 14.3 Ausschnitt aus den Übergangshäufigkeiten der Grapheme (hier Grapheme nach /E/ bezüglich Zahl der /E/ (=100%) und bezüglich Zahl aller Grapheme).
- 14.4 Ausschnitt wie 14.3 in graphischer Form mit logarithmischem Maßstab.

- 15.1 Ausschnitt aus Abstandsmatrix der Grapheme (hier Abstand aller Grapheme von /N/ bezüglich Nachbereich, Vorbereich und Umgebung) in BODL.
- 15.2 Abstandsmatrix für Grapheme der Klasse 1 (gerundet) in BODL.
- 16. Ausschnitt aus der Klassenbildung in BODL für einen Schwellwert und zugehörige Verteilung der n-Tupel.
- 17.1 Ausschnitt aus Wortformen, aufgespaltet in Segmente (BODL).
- 17.2 Ausschnitt aus Wortkonstituenten mit Belegwortform, sortiert (BODL).
- 17.3 Ausschnitt aus Wortkonstituenten, aufgespaltet in 2-Radikal, Kern und 1-Radikal, sortiert (BODL) mit Beleg.
- 17.4 Ausschnitt aus 2-Radikalen, sortiert mit Beleg (BODL).
- 17.5 Ausschnitt aus Kernen, sortiert mit Beleg (BODL).
- 17.6 Ausschnitt aus 1-Radikal, sortiert mit Beleg (BODL).
- 17.7 Liste der Wortkonstituenten-Übergänge, sortiert mit Beleg (BODL).
- 17.8 Liste der Wortformen mit Original- und Graphem-schreibung, sortiert nach der Graphem-schreibung (BODL).
- 17.9 Liste der Wortformen mit Wortkonstituentenstruktur, sortiert nach Strukturmerkmalen (BODL).
- 18.1 Textkennwerte des Texts BODL.
- 18.2 Verteilung der Wortkonstituentenstrukturen (BODL).

---

<sup>1</sup> Alle Beispiel-Ausgaben wurden auf DIN-Format umformatiert, da im Original-Output die volle Papierbreite des Druckerpapiers ausgenützt wurde.

## 6. B i b l i o g r a f h i e

- Abercrombie, D.  
What is a "Letter"?  
Lingua 2, 1949, S.54-63.
- Adamus, M.  
Phonemtheorie und das deutsche Phoneminventar.  
Zur Typologie der germanischen Sprachen.  
Prace Wroclawskiego Towarzystwa Naukowego, Ser.A.,  
Nr. 103, Wroclaw 1967.
- Allén, St.  
Grafematisk analys som grundval för textedering  
med särskild hänsyn till Johan Ekeblads brev till  
brodern Claes Ekeblad 1639-1655.  
Göteborg 1965 (Nordisca Gothoburgensia 1).
- Apresjan, J.D.  
Ideen und Methoden der modernen strukturellen  
Linguistik.  
Kurzer Abriß.  
München 1971.
- Arens, H.  
Verborgene Ordnung.  
Die Beziehungen zwischen Satzlänge und Wortlänge  
in deutscher Erzählprosa vom Barock bis heute.  
Düsseldorf 1965 (Beihefte zu WW 11).
- Arnold, F.G.  
A phonological approach to vowel, consonant and  
syllable in modern French.  
Lingua 5, 1955/56, S.253-287.
- Avram, A.  
Sur quelques particularités des systèmes graphé-  
matiques.  
CLTA 1, 1962, S.9-16.
- Bach, E.  
An Introduction to Transformational Grammars.  
New York-Chicago-San Francisco 1966.
- Bazell, C.E.  
The Grapheme.  
Litera 3, 1956, S.43-46.
- van den Berg, B.  
Diachronisch en synchronisch taalonderzoek.  
In: Handelingen van het 27. Nederlands filologen-  
congres, Groningen 1962, S.33-38.
- Besch, W.  
Schriftzeichen und Laut.  
Möglichkeiten der Lautwertbestimmung an dt.Hand-  
schriften des späten Mittelalters.  
ZfdPh 80, 1961, S.287-302.

- Billmeier, G., Krallmann, D.  
Bibliographie zur statistischen Linguistik.  
Hamburg 1969  
(IPK-F 69-3).
- Bloch, B.  
A set of postulates for phonemic analysis.  
Lg 24, 1948, S. 3-46.
- Bloomfield, L.  
Language.  
London 1965<sup>8</sup>.
- Böhm, S., Koller, G., Schönhut, J., Straßner, E.  
Komplexität von Sätzen als Kriterium für eine  
vergleichende Textanalyse.  
Demnächst in: Linguistik und Didaktik.
- Bolinger, D.L.  
Visual Morphemes.  
Lg 22, 1946, S.333-340.
- Bolinger, D.L.  
Intonation: Levels versus Configurations.  
Word 7, 1951, S.199-210.
- Borgström, C.H.  
The technique of linguistic descriptions.  
ALH 5, 1945-49, S.1-14.
- Bourbaki, N.  
Die Architektur der Mathematik.  
Physikalische Blätter Jg. 17, 1961.  
Heft 4, S.161-166.  
Heft 5, S.212-218.
- Bourne, C.P., Ford, D.F.  
A Study of the Statistics of Letters in English  
Words.  
IC 4 (1961), S.48-67.
- Brekke, H.E.  
Einige Bemerkungen zur Graphematik-Diskussion.  
LB 16, 1971, S.53-59.
- Bünting, K.-D.  
Morphologische Strukturen deutscher Wörter:  
Ein Problem der linguistischen Datenverarbeitung.  
Diss. Bonn 1969.
- Bürmann, G., Frank, H., Lorenz, L.  
Informationstheoretische Untersuchungen über Rang  
und Länge deutscher Wörter.  
GrKG 4, 1963, S.73-90.

- Cohen, M.  
Sur la statistique linguistique.  
Conférences de l'Institut linguistique de Paris,  
9. Année, 1949.  
Paris 1950.
- Colin, A.J.T.  
The Automatic Construction of a Glossary.  
IC 3, 1960, S.211-230.
- Collinder, B.  
Das Wort als phonetische Einheit.  
Uppsala Universitets Årsskrift 13, 1939, S.63-75.
- Crossland, R.A.  
Graphic Linguistics and its Terminology.  
MT 3, 1956, S.8-11.
- Daunt, M.  
Some notes on Old English phonology.  
TPS 1952, 1953, S.48-54.
- DIN 66001 Informationsverarbeitung.  
Sinnbilder für Datenfluß- und Programmlaufpläne.  
1966.
- Dolby, J.L., Resnikoff, H.L.  
The Nature of Affixing in Written English.  
MT 9. Nr. 1, 1966, S.84-89 und Nr. 2, 1966, S.23-33.
- Dolby, J.L., Resnikoff, H.L.  
On the structure of written English words.  
Lg 40, 1964, S.167-197.
- Dolezel, L., Průcha, J.  
A statistical Law of Grapheme Combinations.  
PSML 1, 1966, S.33-43.
- Edgerton, W.F.  
Ideograms in English writing.  
Lg 17, 1941, S.148-150.
- Eggers, H.  
Deutsche Sprachgeschichte III.  
Das Frühneuhochdeutsche.  
Reinbek b. Hamburg 1969 (rororo 270/71).
- Eggers, H. u.a.  
Elektronische Syntaxanalyse der deutschen Gegen-  
wartssprache.  
Tübingen 1969.
- Ellenberg, S.  
Zur Lautstruktur des Wortes.  
Ein kybernetisches Problem.  
ZPhon 17, 1964, S.21-24.
- Endres, C.  
MIX-FORTRAN.  
Vervielfältigte Beschreibung.  
Demnächst in: Mitteilungsblatt des Rechenzentrums  
der Universität Erlangen-Nürnberg.

- Von Essen, O.  
Die Silbe - ein phonologischer Begriff.  
ZPhon 5, 1951, S.199-203.
- von Essen, O.  
Über den Begriff der Silbe.  
WZUB 5, Nr. 1, 1955/56, S.85-88.
- Esser, W.M.  
Zum konsonantischen Element im deutschen Sprach-  
lautkörper.  
Ein Beitrag zur vergleichenden Stilistik.  
WW 10, 1960, S.68-78.
- Feuser, M.  
Der englische Einsilber.  
Diss. Bonn 1948.
- Fischer, W.L.  
Äquivalenz- und Toleranzstrukturen in der Linguistik -  
Zur Theorie der Synonyma.  
Im Druck.
- Fischer, W.L.  
Toleranzräume.  
Archimedes, 22. Jg., Heft IV/1970, S.101-107.
- Fischer, W.L.  
Beispiele für topologische Stilcharakteristiken  
von Texten.  
GrKG 11, 1970, Heft 1, S.1-11.
- Fischer W.L.  
Topologische Stilcharakteristiken von Texten.  
GrKG 10, 1969, Heft 4, S.111-119.
- Fischer, W.L.  
Texte als simpliziale Komplexe.  
BLI 17, 1969, S.27-48.
- Fischer, W.L.  
Zur Algebra der Texte I.  
BLI 7, 1965, S.62-73.
- Fischer, W.L.  
Zur Algebra der Texte II.  
BLI 8, 1966, S.49-64.
- Fleischer, W.  
Strukturelle Untersuchungen zur Geschichte des  
Neuhochdeutschen.  
(Sitzungsber.Sächs.Akad.d.Wiss.Leipzig).  
Berlin 1966.
- Fleischer, W.  
Zum Meißnischen Deutsch.  
Schriftzeichen und Laut.  
PBB 89, Halle 1967, S.58-72.

- Fleischer, W.  
Zum Verhältnis von Phonem und Graphem bei der  
Herausbildung der mhd. Schriftsprache.  
WZUJ 14, 1965, Heft 3, S.461-65.
- Frank, H.  
Informationswissenschaftliche Gesetze in der  
natürlichen deutschen Sprache.  
In: Frank, H. (Hg.)  
Kybernetik - Brücke zwischen den Wissenschaften.  
Frankfurt/Main 1965<sup>5</sup>, S.235-245.
- Frank, H., Klugmann, D., Wendt, S.  
Über den Informationsgehalt der Laute in der  
deutschen Sprache.  
GrKG 4, 1963, S.65-72.
- Freudenberg, R.  
Das Phonem und seine Struktur.  
ZMF 33, 1966, S.1-14.
- Fucks, W.  
Mathematische Analyse des literarischen Stils.  
SG 6, 1953, S.506-523.
- Fucks, W.  
Mathematische Analyse von Sprachelementen, Sprach-  
stil und Sprachen.  
Köln-Opladen 1955  
(Arbeitsgemeinschaft f.Forschung d.Landes Nordrhein-  
Westfalen 34 a).
- Fucks, W.  
Die mathematischen Gesetze der Bildung von Sprach-  
elementen aus ihren Bestandteilen.  
Theorie der Wortbildung aus Silben und der Silben-  
bildung aus Lauten.  
NTF 3 (1956) S.7-21.
- Fucks, W.  
Theorie der Wortbildung.  
Mathematisch-Physikalische Semesterberichte zur  
Pflege des Zusammenhangs von Schule und Universität 4,  
Göttingen 1955, S.195-212.
- Fucks, W.  
Zur Deutung einfachster mathematischer Sprachcharakteri-  
stiken.  
Forschungsberichte des Wirtschafts- und Verkehrs-  
ministeriums Nordrhein-Westfalen, Nr. 344.  
Köln-Opladen 1956.
- Fucks, W., Lauter, J.  
Mathematische Analyse des literarischen Stils.  
In: Gunzenhäuser, R., Kreuzer, H. (Hg.)  
Mathematik und Dichtung.  
Versuch einer exakten Literaturwissenschaft.  
München 1965, S.107-122.

- Gajić, D.  
Der Mehrsilber im Serbo-Kroatischen.  
Diss. Bonn 1950.
- Garvin, P.L.  
Computer participation in linguistic research.  
Lg 38, 1962, S.385-389.
- Garvin, P.L., Spolsky, B. (Hg.)  
Computation in Linguistics.  
A Case Book.  
Bloomington-London 1966.
- Geigy, I.R. (Hg.)  
Wissenschaftliche Tabellen.  
Biberach 1955.
- Grosse, R.  
Das phonematische und das orthographische System  
in der deutschen Gegenwartssprache.  
GW 11, 1967, S.119-129.
- Guiraud, P., Whatmough, J.  
Bibliographie de la statistique linguistique.  
Utrecht-Anvers 1954.
- Guiraud, P.  
Problèmes et méthodes de la statistique linguistique.  
Dordrecht 1959.
- Gunzenhäuser, R.  
Zur mathematischen Textanalyse: Der Vergleich von  
Texten.  
GrKG 1, 1960, S.153-159.
- Hakkarainen, H.J.  
Graphematik und Philologie.  
LiLi 1/2, 1971, S.191-204.
- Halle, M.  
The Strategy of Phonemics.  
Word 10, 1954, S.197-209.
- Hammarström, G.  
Linguistische Einheiten im Rahmen der modernen  
Sprachwissenschaft.  
Berlin-Heidelberg-New York 1966.  
(Kommunikation und Kybernetik in Einzeldarstellungen 5).
- Hamp, E.P.  
Graphemics and Paragraphemics.  
SIL 14, No. 1-2, 1959, S.1-5.

- Hartmann, P.  
Linguistik und maschinelle Datenverarbeitung.  
In: Detering, K., Pilch, H. (Hg.)  
Syntax und Datenverarbeitung II.  
Kolloquium Oberwolfach 28.-29. Juni 1965.  
DFG Forschungsberichte 8.  
Wiesbaden 1966, S.13-38.
- Hartmann, R.  
Bedeutung und Entwicklung der Worthäufigkeitsforschung.  
BLI 4, 1964, S.47-57.
- Harweg, R.  
Buchstabe und Graphem.  
LB 13, 1971, S.78-80.
- Haugen, E.  
The Syllable in Linguistic Description.  
In: Halle, M. (Hg.)  
For Roman Jakobson.  
Essays on the occasion of his sixtieth birthday.  
The Hague 1956, S.213-221.
- Hays, D.G.  
Applied computational linguistics.  
In: Perren, G.E., Trim, J.L.M. (Hg.)  
Applications of Linguistics.  
Selected papers of the Second Int.Congress of  
Applied Linguistics, Cambridge 1969.  
Cambridge 1971, S.65-84.
- Hays, D.G.  
Introduction to Computational Linguistics.  
New York 1967.  
(Mathematical Linguistics and Automatic Language  
Processing, 2).
- Hays, D.G.  
An Introduction to Computational Procedures in  
Linguistic Research.  
In: Ghizetti, A. (Hg.)  
Automatic Translation of Languages.  
Papers presented at NATO Summer School held in  
Venice, July 1962.  
Oxford-London 1966, S.139-165.
- von Held, W.  
Eine verallgemeinerte Phrasenstrukturgrammatik mit  
einem Satzanalyse-Algorithmus.  
IPK-F 33, 1971.

- Henke, H.  
Untersuchungen über das maschinelle Erkennen  
flektierter Wortformen im Deutschen.  
Arbeitsblätter des Lehrstuhls für elektronische  
Rechenanlagen der TH Hannover, Prof.Dr.W.Händler,  
Nr. 13, 1965.
- Herdan, G.  
Type-Token Mathematics.  
A Textbook of mathematical linguistics.  
's Gravenhage 1960.  
(Janua Linguarum, Ser.Maj. 4).
- Herdan, G.  
Quantitative Linguistics.  
London 1964.
- Herdan, G.  
The Calculus of Linguistic Observations.  
's Gravenhage 1962.  
(Janua Linguarum, Ser.Maj. 9).
- Herdan, G.  
Vocabulary Statistics and Phonology: A Parallel.  
Lg 37, 1961, S.247-255.
- Hintze, F.  
Bemerkungen zur Methodik phonologischer Unter-  
suchungen der Wortstruktur.  
SL 2, 1948, S.37-47.
- Hjelmslev, L.  
Prolegomena to a Theory of Language.  
Madison 1961.
- Hjelmslev, L.  
The syllable as a structural unit.  
Proceedings of the Third International Congress  
of Phonetic Science.  
Gand 1938, S.266-272.
- Hockett, C.F.  
Linguistics elements and their relations.  
Lg 37 (1961), S.29-53.
- Hockett, C.F.  
Language, Mathematics and Linguistics.  
The Hague-Paris 1967.
- Jakobson, R. (Hg.)  
The structure of language and its mathematical  
aspects.  
Providence 1961.  
(Proceedings of Symposia in applied Mathematics  
XII).

- Janda, J.W., Rothkegel, A., Zimmermann, H.  
Dokumentation eines Programms zur Analyse  
russischer Sätze.  
Teil 1: BLI 19, 1970, S.7-39.  
Teil 2: BLI 20, 1970, S.7-31.
- Jones, D.  
The Phoneme.  
Its nature and use.  
Cambridge 1950.
- Kettemann, B.  
Welche Lösungsfunktion hat eine graphematische  
Theorie?  
LB 16, 1971, S.52.
- Klötzer, H.  
Gestaltanalyse deutscher und englischer Stamm-  
wörter.  
Ein Beitrag zur Phonologie beider Sprachen.  
Lexis 3, 1953, S.102-108.
- Kloster, J.M.  
Die Silbe in der Phonetik und Phonemik.  
Phonetica 9, 1963, S.17-38.
- Kochendörfer, D.  
Theoretische Grundlagen für die Technifizierung  
der recensio.  
Referat gehalten auf dem Symposium über Probleme  
der maschinellen Verarbeitung mittelhochdeutscher  
Texte, Institut für deutsche Sprache, Mannheim,  
11.-12.6.1971.  
Noch nicht erschienen.
- Krallmann, D.  
Statistische Methoden in der stilistischen Text-  
analyse.  
Ein Beitrag zur Informationserschließung mit Hilfe  
elektronischer Rechenmaschinen.  
(Diss.) Bonn 1966.
- Krámský, J.  
A Quantitative Phonemic Analysis of Italian  
Mono-, Di- und Trisyllabic Words.  
TLP 1, 1964, S.129-144.
- Krámský, J.  
A Quantitative Typology of Languages.  
Language and Speech 2, 1959, S.75-85.
- Küpfmüller, K.  
Die Entropie der deutschen Sprache.  
FTZ 7, 1954, S.265-272.

- Kučera, H.  
Mechanical Phonemic Transcription and Phoneme  
Count of Czech.  
IJSLP 6, 1963, S.36-50.
- Kučera, H., Monroe, G.K.  
A Comparative Quantitative Phonology of Russian,  
Czech and German.  
New York 1968.  
(Mathematical Linguistics and Automatic Language  
Processing 4).
- Kuryłowicz, J.  
Contribution à la théorie de la syllable.  
In: Esquisses Linguistiques.  
Wrocław-Kraków 1960, S.193-220.  
(Prace Językoznawcze 19).
- Lauter, J.  
Information der Sprache - über informationstheo-  
retische Untersuchungen zur deutschen Schrift-  
sprache.  
LiLi 1, 1971, Heft 4, S.9-21.
- Lauter, J.  
Untersuchungen zur Sprache von Kants "Kritik der  
reinen Vernunft".  
Köln-Opladen 1966.  
(Forschungsberichte des Landes Nordrhein-Westfalen  
Nr. 1730).
- Lenders, W.  
Verfahren zur automatischen Herstellung klassifi-  
zierender Indices.  
In: Schanze, H. (Hg.)  
Literatur und Datenverarbeitung.  
Ein Tagungsbericht.  
Tübingen 1972, S.45-55.
- Lenders, W.  
Lexikographische Arbeiten zu Texten der älteren  
deutschen Literatur mit Hilfe von Datenverarbeitungs-  
anlagen.  
ZfdPh 90, 1971, S.321-336.
- Ludviková, M., Königová, M.  
Quantitative Research of Graphemes and Phonemes  
in Czech.  
PBML 7, 1967, S. 15-29.
- Lutz, H.D.  
Probleme der grammatischen Beschreibung an spät-  
mittelhochdeutschen Texten.  
In: Schanze, H. (Hg.)  
Literatur und Datenverarbeitung.  
Ein Tagungsbericht.  
Tübingen 1972, S.56-65.

- Lutz, H.D.  
Übersicht zur maschinellen Analyse altdeutscher  
Texte.  
ZfdPh 90, 1971, S.336-355.
- Malmberg, B.  
Structural Linguistics and Human Communication.  
Berlin-Göttingen-Heidelberg 1963.
- Mandelbrot, B.  
Structure formelle des Textes et Communication:  
deux études.  
Word 10, 1954, S.1-27.  
Addenda à l'article "Structure formelle des textes  
et Communication".  
Word 11, 1955, S.424-425.
- McIntosh, A.  
The Analysis of Written Middle English.  
TPS 1956, S.26-55.
- McLaughlin, J.C.  
A Graphemic-Phonemic Study of a Middle English  
Manuscript.  
The Hague 1963.
- Meier, H.  
Deutsche Sprachstatistik.  
2 Bände.  
Hildesheim 1964
- Menzerath, P.  
Die Architektonik des deutschen Wortschatzes.  
Bonn-Hannover-Stuttgart 1954.  
(Phonetische Studien 3).
- Menzerath, P.  
Zum Aufbau des deutschen Wortschatzes.  
Forschungen und Fortschritte 20, 1944, S.76-77.
- Menzerath, P., Meyer-Eppler, W.  
Sprachtypologische Untersuchung.  
SL 4, 1950, S.54-93.
- Meyer-Eppler, W.  
Grundlagen und Anwendungen der Informationstheorie.  
Berlin-Heidelberg-New York 1959, 1969<sup>2</sup> (bearb. von  
G.Heike und K.Löhn).  
(Kommunikation und Kybernetik in Einzeldarstellungen  
1).
- Meyer-Eppler, W.  
Zur Anwendung informationstheoretischer Methoden  
auf sprachliche Probleme.  
ZPhon 8, 1954, S.32-40.

- Michel, W.-D.  
Die graphemische Entwicklung der s-Laute im Deutschen.  
PBB 81, Halle 1959, S.456-480.
- Moulton, W.G.  
Syllable Nuclei and Final Consonant Clusters in German.  
In: Halle, M. (Hg.)  
For Roman Jakobson.  
Essays on the occasion of his sixtieth birthday.  
Den Haag 1956, S.372-381.
- Moulton, W.G.  
Phonemische Segmentierungsmerkmale in der deutschen Hochsprache der Gegenwart.  
In: Steger, H. (Hg.)  
Vorschläge für eine strukturelle Grammatik des Deutschen.  
Darmstadt 1970, S.429-453.
- Moulton, W.G.  
The Vowels of Dutch: Phonetic and distributional Classes.  
Lingua 11, 1962, S.294-312.
- Müller, Ch.  
Initiation á la Statistique Linguistique.  
Paris 1968.
- Must, G.  
The Symbols for /i/ and /j/ in German Orthography:  
an Historical Survey.  
MLN 80, 1965, S.584-595.
- Naur, P. (Hg.)  
Revised Report on the Algorithmic Language ALGOL 60.  
Numerische Mathematik 4, 1963, S.420-453.
- O'Connor, J.D., Trim, J.L.  
Vowel, Consonant and syllable - a phonological definition.  
Word 9, 1953, S.103-122.
- Penttilä, A.  
Zur Grundlagenforschung der geschriebenen Sprache.  
ASLU 2,2, 1970.
- Penttilä, A.  
Zur Grundlagenforschung der geschriebenen Sprache.  
LB 16, 1971, S.49-52.
- Penttilä, A.  
Einige Bemerkungen über die Unterscheidung von Sprache und Rede.  
Actes du quatrième Congrès international de Linguistes.  
Copenhagen 1938, S.157-163.

- Piirainen, I.T.  
Graphematische Untersuchungen zum Frühneuhochdeutschen.  
Berlin 1968.
- Piirainen, I.T.  
Grapheme als quantitative Größen.  
LB 13, 1971, S.81-82.
- Piirainen, I.T.  
Historische Graphematik und die datenverarbeitenden  
Methoden.  
FoL 3, 1969, S.145-152.
- Piirainen, I.T.  
Quantitative Methoden in der Sprachforschung.  
LB 3, 1969, S.22-30.
- Piirainen, I.T.  
Quantitative Analyse der Graphem- und Phonemebene.  
PBML 11, 1969, S.3-15.
- Piirainen, I.T.  
Quantitative Vorgehensweisen in der automatischen  
Analyse der älteren deutschen Trivialliteratur.  
In: Schanze, H. (Hg.)  
Literatur und Datenverarbeitung.  
Ein Tagungsbericht.  
Tübingen 1972, S.88-106.
- Piirainen, I.T., Wenzel, G.  
Contribution to the Foundation of Automatic Text  
Processing. (Documentation)  
IBM Publication 71.11.006, 1971.
- Plath, W.  
Mathematical Linguistics.  
In: Mohrmann, Chr., Sommerfeldt, A., Whatmough, J. (Hg.)  
Trends in European and American Linguistics 1930-1960.  
Utrecht-Antwerpen 1961, S.21-57.
- Posner, R.  
The use and abuse of stylistic statistics.  
ArchL 15, 1964, S.111-139.
- Pulgram, E.  
Phoneme and Grapheme: A Parallel.  
Word 7, 1951, S.15-20.
- Pulgram, E.  
Syllable, Word, Nexus, Cursus.  
The Hague 1970.  
(Janua Linguarum, Ser.Min.81).

- Pulgram, E.  
Graphic and Phonic Systems: Figurae and Signs.  
Word 21, 1965, S.208-224.
- Pulgram, E.  
Consonant Cluster, Consonant Sequence and the Syllable.  
Phonetica 13, 1965, S.76-81.
- Rath, R.  
Vorschläge zur Automatischen Lemmatisierung (AL) deutscher Adjektive.  
LB 12, 1971, S.53-59.
- Rosetti, A.  
Sur la théorie de la syllable.  
s'Gravenhage 1959.  
(Janua Linguarum 9).
- Rosić, M.  
Der Einsilber im Serbo-kroatischen.  
(Diss.) Bonn 1950.
- Samuels, M.L.  
The study of Old English Phonology.  
TPS 1952, S.15-47.
- Schmitt, L.E. (Hg.)  
Deutsche Wortindices.  
Berlin 1970ff (bisher 3 Bände).
- Schnelle, H.  
Methoden mathematischer Linguistik.  
In: Enzyklopädie der geisteswissenschaftlichen Arbeitsmethoden.  
4. Methoden der Sprachwissenschaft.  
München-Wien 1968, S.135-160.
- Schnelle, H.  
Mathematische Linguistik - Eine kurze Einführung.  
ZMF 33, 1966, S.193-206.
- Schnelle, H., Kranzhoff, J.A.  
Zur Beschreibung und Bearbeitung der Struktur deutscher Wörter; Bemerkungen zum sprachtheoretischen Rahmen.  
1. Teil: BLI 5, 1965, S.80-89.  
2. Teil: BLI 6, 1965, S.65-87.
- Schönhut, J.  
Die Verwendung von Compiler-Compilern zur Programmierung von Analysegrammatiken.  
Referat gehalten am 12.11.1970 beim Kolloquium "Programmiersprachen für Linguistische Datenverarbeitung" beim Institut für deutsche Sprache, Mannheim.
- Schönhut, J.  
Kurzbeschreibung des Flußdiagramm-Compilers FLOW.  
Interne Programmbeschreibung, Erlangen 1970.  
Demnächst in: Mitteilungsblatt des Rechenzentrums der Universität Erlangen-Nürnberg.

- Schönhut, J.  
Kurzbeschreibung eines Programmsystems zur Auswertung  
syntaktischer Beschreibungen auf der CD 3300/MASTER.  
Interne Programmbeschreibung.
- Schönhut, J.  
Ein komfortables Dokumentationssystem für FORTRAN-  
Programme.  
Demnächst in: Mitteilungsblatt des Rechenzentrums  
der Universität Erlangen-Nürnberg.
- Schönpflug, W.  
n-Gramm-Häufigkeiten in der deutschen Sprache.  
I. Monogramme und Diagramme.  
Zeitschrift für experimentelle und angewandte  
Psychologie 16, 1969, S.157-183.
- Schröder, W.  
Der Computer in der deutschen Philologie des  
Mittelalters.  
PBB 91, Tübingen 1969, S.386-396.
- Schwerte, H., Schanze, H. (Hg.)  
Indices zur deutschen Literatur.  
Frankfurt/Main 1968ff (bisher 7 Bände).
- Sckommodau, H.  
Die Silbe und die Struktur des Französischen.  
Wiesbaden 1967.  
(Sitzungsberichte d. Wiss. Ges. a. d. Johann-Wolfgang-  
Goethe-Universität, Frankfurt/M., 1966, Nr. 4).
- Seiler, H.  
Laut und Sinn: Zur Struktur der deutschen Einsilber.  
Lingua 11, 1962, S.375-385.
- Seiler, H.  
On defining the word.  
Proceedings of the 9th International Congress of  
Linguistics.  
The Hague 1964, S.767-770.
- Simon, G.  
Zur Theorie der Kontamination.  
Referat gehalten auf dem Symposium über Probleme  
der maschinellen Verarbeitung mittelhochdeutscher  
Texte, Institut für deutsche Sprache, Mannheim,  
11.-12.6.1971.  
Noch nicht erschienen.
- Simon, G.  
Die erste deutsche Fastnachtsspieltradition.  
Zur Überlieferung, Textkritik und Chronologie der  
Nürnberger Fastnachtsspiele des 15. Jahrhunderts  
(mit kurzen Einführungen im Verfahren der quanti-  
tativen Linguistik).  
Lübeck-Hamburg 1970.  
(Germanistische Studien 240).

- Singer, H.  
Der Graphembegriff bei der Analyse altdeutscher  
Handschriften.  
LB 13, 1971, S.83-85.
- Skalička, V.  
Konsonantenkombinationen und linguistische Typologie.  
TLP 1, 1964, S.111-114.
- Spang-Hanssen, H.  
Probability and Structural Classification in  
Language Description.  
Copenhagen 1959.
- Stetson, R.H.  
The Phoneme and the Grapheme.  
Mélanges de Linguistiques et de Philologie offerts  
à Jacq. van Ginneken.  
Paris 1937, S.353-356.
- Straßner, E.  
Klassenbildung auf der Graphemebene und Wort-  
konstituenz.  
Habil-Schrift in Vorbereitung.
- Straßner, E., Schönhut, J.  
Sozio- und psycholinguistische Aspekte der  
Nachrichtensprache.  
Vortrag, gehalten am 25.9.1971 in Baden-Baden.  
Abgedruckt in: Materialien zur Arbeitstagung  
"Sozio- und psycholinguistische Aspekte der  
Nachrichtensprache" veranstaltet von der Fachgruppe  
"Rundfunk und Sprache" in Zusammenarbeit mit der  
Hörfunkdirektion des Südwestfunks, Erlangen 1972.  
Eine überarbeitete Fassung dieses Vortrags demnächst  
in dem Sammelband "Sprache und Gesellschaft",  
(Fink-Verlag), München 1972.
- Straßner, E., Schönhut, J.  
Probleme einer Klassenbildung bei der graphemati-  
schen Analyse.  
Referat gehalten auf dem Symposium über Probleme  
der maschinellen Verarbeitung mittelhochdeutscher  
Texte, Institut für deutsche Sprache, Mannheim,  
11.-12.6.1971.  
Noch nicht erschienen.
- Tillmann, H.G.  
Das phonetische Silbenproblem - Eine theoretische  
Untersuchung.  
(Diss.) Bonn 1964.
- de Tollenaere, F.  
Die Automatisierung in der Lexikographie.  
BLI 1, 1963, S.33-44.

- de Tollenaere, F.  
Nieuwe Wege in de Lexicologie.  
Amsterdam 1963.
- Uldall, H.J.  
Speech and writing.  
ALH 4, 1944, S.11-16.
- Ulverstad, B.  
Statistik und Sprachbeschreibung.  
In: Moser, H. (Hg.)  
Das Ringen um eine neue deutsche Grammatik.  
Darmstadt 1962, S.61-73.
- Ungeheuer, G.  
Linguistische Datenverarbeitung - die Realität  
und eine Konzeption.  
IBM-Nachrichten 21, 1971, S.688-694.
- Ungeheuer, G., Kaestner, W.  
Untersuchung zur Transformation deutscher Schrift-  
texte in entsprechende Phonemtexte mit Hilfe elek-  
tronischer Rechenmaschinen.  
IPK-F 66/1, 1966.
- USA Standard FORTRAN  
USAS X3.9-1966.  
New York 1966.
- Vachek, J.  
Zum Problem der geschriebenen Sprache.  
In: Vachek, J. (Hg.)  
A Prague School Reader in Linguistics.  
Bloomington-London 1966<sup>2</sup>, S.441-452.
- Vachek, J.  
Written Language and Printed Language.  
In: Vachek, J. (Hg.)  
Prague School Reader in<sup>2</sup> Linguistics.  
Bloomington-London 1966<sup>2</sup>, S.453-460.
- Vachek, J.  
Some remarks on writing and phonetic transcription.  
ALH 5, 1945-49, S.86-93.
- Vogt, H.  
Phoneme Classes and Phoneme Classification.  
Word 10, 1954, S.28-34.
- Weinacht, H.  
Das Ambraser Heldenbuch (Arbeitstitel Diss.).  
In Vorbereitung.
- Wenzel, G.  
Beitrag zur Definition der Grapheme.  
LB 13, 1971, S.80-81.

- Wenzel, G.  
Textverarbeitung auf der Graphemebene.  
(Diss.) Stuttgart 1967.
- Wis, M.  
Fremdes SK im Frühneuhochdeutschen.  
NphM 65, 1964, S.54-70.
- Wisbey, R.  
Der Computer im Dienst der Literaturwissenschaft.  
In: Schanze, H. (Hg.)  
Literatur und Datenverarbeitung.  
Ein Tagungsbericht.  
Tübingen 1972, S.8-25.
- Wisbey, R. (Hg.)  
The computer in literary and linguistic research.  
Papers from a Cambridge symposium.  
Cambridge 1971.
- Wisbey, R. (Hg.)  
Compendia, Computer Generated Aids to Literary  
and Linguistics Research.  
Leeds 1968ff (bisher 3 Bände).
- 3100 3200 3300 3500 Computer Systems COMPASS Reference  
Manual.  
Control Data Corporation, Pub.No.60236800A, 1969.
- 3100 3200 3500 Computer Systems USASI/FORTRAN MASTER/MSOS  
Reference Manual.  
Control Data Corporation, Pub.No.60281400, 1970.
- 3170 3300 3500 MASTER Version 3 Installation Handbook.  
Control Data Corporation, Pub.No.60280500G, 1971.
- 3300 3500 Computer Systems MASTER Version 3 Reference  
Manual.  
Control Data Corporation, Pub.No. 60281000, 1970.
- 3300 Computer System Reference Manual.  
Control Data Corporation, Pub.No.60157000, 1968<sup>3</sup>.
- 3300 3500 Computer Systems MASTER Version 3 Reference  
Manual.  
Control Data Corporation, Pub.No.60281000, 1970.
- 3300 3500 Computer Systems Mass Storage SORT/MASTER.  
Control Data Corporation, Pub.No.60206600, 1967.

6.1 Abkürzungsverzeichnis zur Bibliographie

|        |                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------|
| ALH    | Acta Linguistica Acad.Sc.Hungariae, Budapest                                                                  |
| ArchL  | Archivum Linguisticum. A Review of Comparative Philology and General Linguistics, Glasgow.                    |
| ASLU   | Acta Societatis Linguisticae Upsaliensis, N.S., Uppsala                                                       |
| BLI    | Beiträge zur Linguistik und Informationsverarbeitung, München und Wien                                        |
| CLTA   | Cahiers de linguistique théoretique et appliquée, Bukarest                                                    |
| FoL    | Folia Linguistica, Acta Societatis Linguisticae Europaeae, The Hague                                          |
| FTZ    | Fernmeldetechnische Zeitschrift, Braunschweig                                                                 |
| GrKG   | Grundlagenstudien aus Kybernetik und Geisteswissenschaft, Quickborn                                           |
| GW     | Germanica Wratislaviensia, Wrocław                                                                            |
| IC     | Information and Control, New York                                                                             |
| IPK-F  | IPK-Forschungsbericht, Bonn (Institut für Phonetik und Kommunikationsforschung, Universität Bonn)             |
| IJSLP  | International Journal of Slavic Linguistics and Poetics, 's Gravenhage                                        |
| LB     | Linguistische Berichte, Braunschweig                                                                          |
| Lg     | Language, Journal of the Linguistic Society of America, Baltimore                                             |
| LiLi   | LiLi, Zeitschrift für Literaturwissenschaft und Linguistik, Frankfurt/Main                                    |
| Lingua | Lingua, International Review of General Linguistics, Revue internationale de linguistique générale, Amsterdam |
| MLN    | Modern Language Notes, Baltimore                                                                              |
| MT     | Mechanical Translation, Cambridge/Mass.                                                                       |
| NphM   | Neuphilologische Mitteilungen, Bulletin de la Société néophilologique de Helsinki, Helsinki                   |

|           |                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| NTF       | Nachrichtentechnische Fachberichte, Berlin                                                                               |
| PBB       | Beiträge zur Geschichte der deutschen Sprache und Literatur, Tübingen bzw. Halle                                         |
| PBML      | The Prague Bulletin of Mathematical Linguistics, Prag                                                                    |
| Phonetica | Phonetica, Internationale Zeitschrift für Phonetik, International Journal of Phonetics, Basel-New York                   |
| PSML      | Prague Studies in Mathematical Linguistics, Prag                                                                         |
| SG        | Studium Generale, Berlin-Göttingen-Heidelberg                                                                            |
| SIL       | Studies in Linguistics, Buffalo                                                                                          |
| SL        | Studia Linguistica, Revue de linguistique générale et comparée, Lund                                                     |
| TLP       | Travaux linguistiques de Prague, Prag                                                                                    |
| TPS       | Transactions of the Philological Society, Oxford                                                                         |
| Word      | Word, Journal of the Linguistic Circle of New York, New York                                                             |
| WW        | Wirkendes Wort, Deutsches Sprachschaffen in Lehre und Leben, Düsseldorf                                                  |
| WZUB      | Wissenschaftliche Zeitschrift der Humboldt-Universität, Berlin, Gesellschafts- und sprachwissenschaftliche Reihe         |
| WZUJ      | Wissenschaftliche Zeitschrift der Friedrich-Schiller-Universität, Jena, Gesellschafts- und sprachwissenschaftliche Reihe |
| ZMF       | Zeitschrift für Mundartforschung, Wiesbaden                                                                              |
| ZPhon     | Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung, Berlin                                         |



Bisher sind folgende Mitteilungsblätter des Rechenzentrums der Friedrich-Alexander-Universität Erlangen-Nürnberg erschienen:

- Nr. 1 - Allgemeine Information  
Erlangen, November 1968
- Nr. 2 - Jahresbericht 1968  
Erlangen, Februar 1969
- Nr. 3 - J. Kettler  
Über den Umgang mit dem ALGOL-Kompiler der  
CD 3300 unter dem Betriebssystem MASTER  
Erlangen, September 1969
- Nr. 4 - J. Kettler  
Einführung in die ALGOL-Programmierung  
für die Rechenanlage CD 3300  
Erlangen, Januar 1970
- Nr. 5 - Jahresbericht 1969  
Erlangen, Februar 1970
- Nr. 6 - Jahresbericht 1970  
Erlangen, Februar 1971
- Nr. 7 - Heinz Zemanek  
Zukunftsaspekte in der Informationsverarbeitung  
Erlangen, Oktober 1971
- Nr. 8 - Allgemeine Information  
Erlangen, Januar 1972
- Nr. 9 - G. Seybold  
EDI - ein einfacher Dialoginterpreter  
Diplomarbeit  
Erlangen, Januar 1972
- Nr. 10 - Jahresbericht 1971  
Erlangen, Januar 1972
- Nr. 11 - G. Goerz  
LISP 1.5 - Handbuch für die CD 3300  
Erlangen, März 1972





## **Offsetdruck-Fotodruck**

---

J. Hög, 8520 Erlangen Hauptstr. 109, (Nähe Martin-Luther-Platz) Tel. 22990